

# **SISTEM PEMANTAUAN MENGGUNAKAN BLYNK DAN PENGENDALIAN PENYIRAMAN TANAMAN JAMUR DENGAN METODE LOGIKA *FUZZY***

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun oleh:  
Handi  
NIM: 145150301111067



**PROGRAM STUDI TEKNIK KOMPUTER  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018**



## PENGESAHAN

SISTEM PEMANTAUAN MENGGUNAKAN BLYNK DAN PENGENDALIAN  
PENYIRAMAN TANAMAN JAMUR DEGAN METODE LOGIKA FUZZY

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun Oleh:

Handi

NIM: 145150301111067

Skripsi ini telah diuji dan dinyatakan lulus pada  
27 Desember 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Hurriyatul Fitriyah, S.T., M.Sc  
NIP: 19851001 201504 2 003

Dosen Pembimbing II

Gembong Edhi Setyawan, S.T., M.T  
NIK: 201208 761201 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D  
NIP: 19710518 200312 1 001



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 20 Desember 2018



Handi

NIM: 145150301111067

## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan atas rahmat dan karunia Tuhan yang Maha Esa, sehingga penulis dapat menyelesaikan skripsinya dengan judul **"SISTEM PEMANTAUAN MENGGUNAKAN BLYNK DAN PENGENDALIAN PENYIRAMAN TANAMAN JAMUR DENGAN METODE LOGIKA FUZZY"**. ini dapat terselesaikan. Penulis menyadari bahwa pembuatan skripsi ini tidak lepas atas banyaknya dukungan dan bantuan. Maka penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Ibu Hurriyatul Fitriyah, S.T., M.Sc dan Bapak Gembong Edhi Setyawan, S.T., M.T selaku dosen pembimbing yang telah memberikan dukungan, saran, serta ilmu dalam membimbing penulis sehingga pengerjaan skripsi selesai.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Ayahanda dan Ibunda, selaku orang tua penulis yang selalu memberikan motivasi, semangat dan do'a sehingga penulis dapat menyelesaikan skripsi.
4. Ibu Ratih Kusumasari Ndaru, S.P. dan Ibu Sri Zunaini Sa'adah, S.P. selaku pembimbing di Balai Pengkajian Teknologi Pertanian yang telah memberikan nasihat dan saran kepada penulis.
5. Segenap dosen dan staf Fakultas Ilmu Komputer Universitas Brawijaya atas ilmu pengetahuan serta nasihat dan bantuan administratif yang diberikan.

Dengan segala keterbatasan pengetahuan yang dimiliki, penulis menyadari bahwa laporan ini masih banyak kekurangan, oleh sebab itu penulis berharap adanya pengembangan lebih lanjut oleh pihak-pihak terkait. Semoga skripsi ini dapat bermanfaat bagi penulis dan bagi semua pihak yang menggunakannya.

Malang, 13 Desember 2018

Penulis

Hanafihandi96@gmail.com



## ABSTRAK

Pemantauan dan penyiraman merupakan hal utama yang harus diperhatikan saat mengelola tanaman jamur. Pemantauan memegang peranan yang sangat penting untuk mengetahui suhu dan kelembapan yang ada di dalam ruangan kubung jamur, sedangkan penyiraman dibutuhkan untuk mengatur kelembapan karena jika terlalu lembab dapat menyebabkan *baglog* jamur pada media tanam menghitam. Berdasarkan permasalahan tersebut perlu adanya penelitian terkait dengan sistem otomatisasi untuk memantau dan mengendalikan penyiraman sesuai dengan kondisi yang dibutuhkan pada kubung jamur. Pada penelitian ini menggunakan aplikasi Blynk untuk pemantauan dari jauh dan sensor DHT11 sebagai parameter untuk mendeteksi suhu dan kelembapan. Metode *Fuzzy Sugeno* dipilih untuk mengendalikan penyiraman pada kubung jamur sesuai kebutuhan dengan menyiram air sebanyak yang sudah ditentukan dari hasil *rule fuzzy*. Berdasarkan hasil pengujian pembacaan sensor DHT11 memiliki rata-rata *error* 4,07%. Pada pengujian sistem kontrol penyiraman yang diujikan sebanyak 10 kali percobaan dengan presentase *error* sebanyak 16,66%. Perbandingan durasi pengujian memiliki perbedaan 45 detik, sedangkan pada jumlah pengujian penyiraman memiliki perbedaan 500 mililiter air. Meskipun pelaksanaan manual lebih banyak dan lama, namun akurasi dari sensor dan penyiraman dengan logika *fuzzy* lebih sesuai dengan keadaan yang dibutuhkan.

Kata kunci: Pemantauan, Jamur tiram, Sensor DHT11, *Fuzzy Sugeno*, Blynk

## ABSTRACT

*Monitoring and watering are the main processes that must be considered when managing mushroom plants. Monitoring plays a very important role in knowing the temperature and humidity in the mushroom contact room, while watering is needed to regulate humidity because if it is too humid it can cause the baglog of fungi to grow in black media. Based on these problems it is necessary to conduct a research related to the automation system for monitoring and controlling watering according to the conditions needed on the mushroom. In this study using the Blynk application for remote monitoring and DHT11 sensors as parameters to detect temperature and humidity. The Fuzzy Sugeno method was chosen to control the watering of the mushroom casing as needed by watering parallelism as determined from the results of the fuzzy rule. Based on the test results the reading of the DHT11 sensor has an average error of 4.07%. In testing the watering control system tested 10 times with an error percentage of 16.66%. Comparison of test duration has a difference of 45 seconds, while the number of watering tests has a difference of 500 milliliters of water. Although manual implementation is lot and longer, the accuracy of the sensor and watering with fuzzy logic is corresponding in accordance with the conditions needed.*

**Keywords:** Monitoring, Oyster Mushrooms, DHT11 Sensors, Fuzzy Sugeno, Blynk



## DAFTAR ISI

SISTEM PEMANTAUAN MENGGUNAKAN BLYNK DAN PENGENDALIAN PENYIRAMAN TANAMAN JAMUR DENGAN METODE LOGIKA <i>FUZZY</i> .....	i
PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN .....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan .....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Kajian Pustaka .....	5
2.2 Dasar Teori .....	6
2.2.1 Jamur Tiram.....	7
2.2.2 Logika <i>Fuzzy</i> .....	8
2.2.3 Mikrokontroler .....	9
2.2.4 Sensor .....	10
2.2.5 Aktuator .....	11
2.2.6 Blynk.....	12
2.2.7 Arduino IDE .....	13
BAB 3 METODOLOGI PENELITIAN .....	15

3.1 Alur Metode Penelitian.....	15
3.2 Studi Literatur .....	15
3.3 Analisis Kebutuhan .....	16
3.4 Perancangan dan Implementasi .....	17
3.4.1 Perancangan.....	17
3.4.2 Implementasi .....	17
3.5 Pengujian dan Analisis .....	17
3.6 Penutup.....	17
<b>BAB 4 ANALISIS KEBUTUHAN .....</b>	<b>18</b>
4.1 Gambaran Umum Sistem.....	18
4.2 Kebutuhan Pengguna.....	18
4.2.1 Kebutuhan Antarmuka .....	19
4.3 Kebutuhan Sistem .....	19
4.3.1 Kebutuhan Perangkat Keras.....	20
4.3.2 Kebutuhan Perangkat Lunak .....	28
4.3.3 Kebutuhan Fungsional.....	28
4.3.4 Kebutuhan Non-fungsional .....	28
4.3.5 Batasan Desain Sistem .....	29
<b>BAB 5 PERANCANGAN DAN IMPLEMENTASI .....</b>	<b>30</b>
5.1 Perancangan Sistem.....	30
5.1.1 Perancangan Perangkat Keras .....	30
5.1.2 Perancangan Perangkat Lunak.....	34
5.2 Implementasi Sistem .....	44
5.2.1 Implementasi Perangkat Keras .....	44
5.2.2 Implementasi Perangkat Lunak.....	47
<b>BAB 6 PENGUJIAN DAN ANALISIS.....</b>	<b>54</b>
6.1 Pengujian Sensor DHT11.....	54
6.1.1 Tujuan Pengujian.....	54
6.1.2 Prosedur Pengujian .....	54
6.1.3 Hasil Pengujian .....	54
6.1.4 Analisis Pengujian.....	55
6.2 Pengujian Pengamatan .....	57



6.2.1 Tujuan Pengujian.....	57
6.2.2 Prosedur Pengujian .....	57
6.2.3 Hasil Pengujian .....	57
6.2.4 Analisis Pengujian.....	58
6.3 Pengujian Metode Logika <i>Fuzzy</i> .....	58
6.3.1 Tujuan Pengujian.....	58
6.3.2 Prosedur Pengujian .....	58
6.3.3 Hasil Pengujian .....	59
6.3.4 Analisis Pengujian.....	60
BAB 7 PENUTUP .....	62
7.1 Kesimpulan.....	62
7.2 Saran .....	62
DAFTAR PUSTAKA.....	64
LAMPIRAN A DATA PENELITIAN .....	68
A.1 Surat Permohonan Penelitian .....	68
A.2 Hasil Wawancara .....	69
A.3 Rules Penyiraman .....	70
LAMPIRAN B KODE PROGRAM.....	71
B.1 Main Program .....	71
B.2 Subprogram Fuzzifikasi .....	73
B.3 Subprogram Inferensi .....	74
B.4 Subprogram Defuzzifikasi .....	74

## DAFTAR TABEL

Tabel 2.1 Kajian pustaka .....	6
Tabel 2.2 Sketsa Arduino IDE .....	14
Tabel 4.1 Spesifikasi NodeMCU .....	21
Tabel 4.2 Spesifikasi sensor DHT11.....	23
Tabel 4.3 Konfigurasi pin DHT11 .....	23
Tabel 4.4 Spesifikasi Relay.....	24
Tabel 4.5 Pin <i>input</i> pada I2C.....	26
Tabel 4.6 Pin <i>input</i> dan <i>output power bank</i> .....	28
Tabel 5.1 Konfiguasi pin sensor DHT11 dengan NodeMCU.....	31
Tabel 5.2 Konfigurasi I2C dengan NodeMCU .....	32
Tabel 5.3 Konfigurasi pin subsistem penyiraman .....	33
Tabel 5.4 <i>Rule fuzzy</i> .....	39
Tabel 5.5 Definisi <i>rule fuzzy</i> .....	42
Tabel 5.6 Kode Program autentikasi token Blynk.....	47
Tabel 5.7 Kode Program <i>setup</i> .....	48
Tabel 5.8 Kode Program <i>loop</i> .....	48
Tabel 5.9 Kode Program fuzzifikasi .....	49
Tabel 5.10 Kode Program inferensi.....	51
Tabel 5.11 Kode Program defuzzifikasi .....	51
Tabel 5.12 Kode Program penyiraman .....	52
Tabel 6.1 Analisis perbandingan pengujian sensor.....	55
Tabel 6.2 Hasil pengujian penyiraman .....	59
Tabel 6.3 Analisis perbandingan pengujian penyiraman .....	60



## DAFTAR GAMBAR

Gambar 2.1 Diagram blok sistem yang dikembangkan Prayitno (2017) .....	5
Gambar 2.2 Diagram blok sistem yang dikembangkan Kaewwiset & Yodkhad (2017) .....	6
Gambar 2.3 Bentuk fisik jamur tiram .....	7
Gambar 2.4 Logika <i>fuzzy</i> .....	8
Gambar 2.5 Arsitektur logika <i>fuzzy</i> .....	9
Gambar 2.6 IC mikrokontroler .....	9
Gambar 2.7 Blok diagram mikrokontroler .....	10
Gambar 2.8 Pengukuran suhu pada termometer .....	11
Gambar 2.9 Pengukur kelembapan pada higrometer .....	11
Gambar 2.10 Aktuator .....	12
Gambar 2.11 Antarmuka aplikasi Blynk .....	12
Gambar 2.12 <i>Sketch</i> Arduino IDE .....	13
Gambar 2.13 <i>Board manager</i> Arduino IDE .....	14
Gambar 3.1 Diagram alur metodologi penelitian .....	15
Gambar 4.1 Diagram analisis kebutuhan .....	18
Gambar 4.2 Kebutuhan pengguna .....	19
Gambar 4.3 Diagram blok sistem .....	19
Gambar 4.4 NodeMCU <i>development kit</i> .....	21
Gambar 4.5 Skema pin-pin NodeMCU .....	22
Gambar 4.6 Skematik sensor DHT11 .....	22
Gambar 4.7 Skematik Relay .....	24
Gambar 4.8 Pompa air mini .....	25
Gambar 4.9 Penerapan komponen I2C pada LCD .....	25
Gambar 4.10 Modul <i>step down</i> LM2596 .....	26
Gambar 4.11 <i>Smartphone</i> .....	27
Gambar 4.12 <i>Power bank</i> .....	27
Gambar 5.1 Blok diagram perancangan sistem .....	30
Gambar 5.2 Subsistem sensor DHT11 .....	31
Gambar 5.3 Perancangan subsistem LCD .....	32

Gambar 5.4 Perancangan subsistem penyiraman .....	33
Gambar 5.5 Komponen keseluruhan .....	34
Gambar 5.6 Perancangan diagram alir program sistem .....	35
Gambar 5.7 Perancangan diagram alir metode <i>fuzzy</i> .....	37
Gambar 5.8 Fungsi himpunan suhu .....	37
Gambar 5.9 Fungsi himpunan kelembapan .....	38
Gambar 5.10 Fungsi himpunan suhu .....	40
Gambar 5.11 Fungsi himpunan kelembapan .....	40
Gambar 5.12 <i>Rules evaluation</i> suhu .....	40
Gambar 5.13 <i>Rule evaluation</i> kelembapan.....	41
Gambar 5.14 Fungsi <i>singleton</i> durasi penyiraman .....	43
Gambar 5.15 Perancangan <i>widget</i> Blynk.....	44
Gambar 5.16 <i>Email</i> menerima autentikasi token .....	44
Gambar 5.17 Implementasi hubungan NodeMCU dengan sensor dan aktuator .	44
Gambar 5.18 Implementasi <i>protoype</i> antarmuka .....	45
Gambar 5.19 Implementasi subsistem sensor DHT11 .....	45
Gambar 5.20 Implementasi subsistem komunikasi NodeMCU dengan Blynk .....	46
Gambar 5.21 Implementasi subsistem penyiraman .....	46
Gambar 5.22 Implementasi subsistem komponen keseluruhan .....	47
Gambar 6.1 Hasil pengujian sensor .....	54
Gambar 6.2 Hasil perbandingan pengujian alat ukur .....	57



DAFTAR LAMPIRAN

LAMPIRAN A DATA PENELITIAN ..... 68

    A.1 Surat Permohonan Penelitian ..... 68

    A.2 Hasil Wawancara ..... 69

    A.3 Rules Penyiraman ..... 70

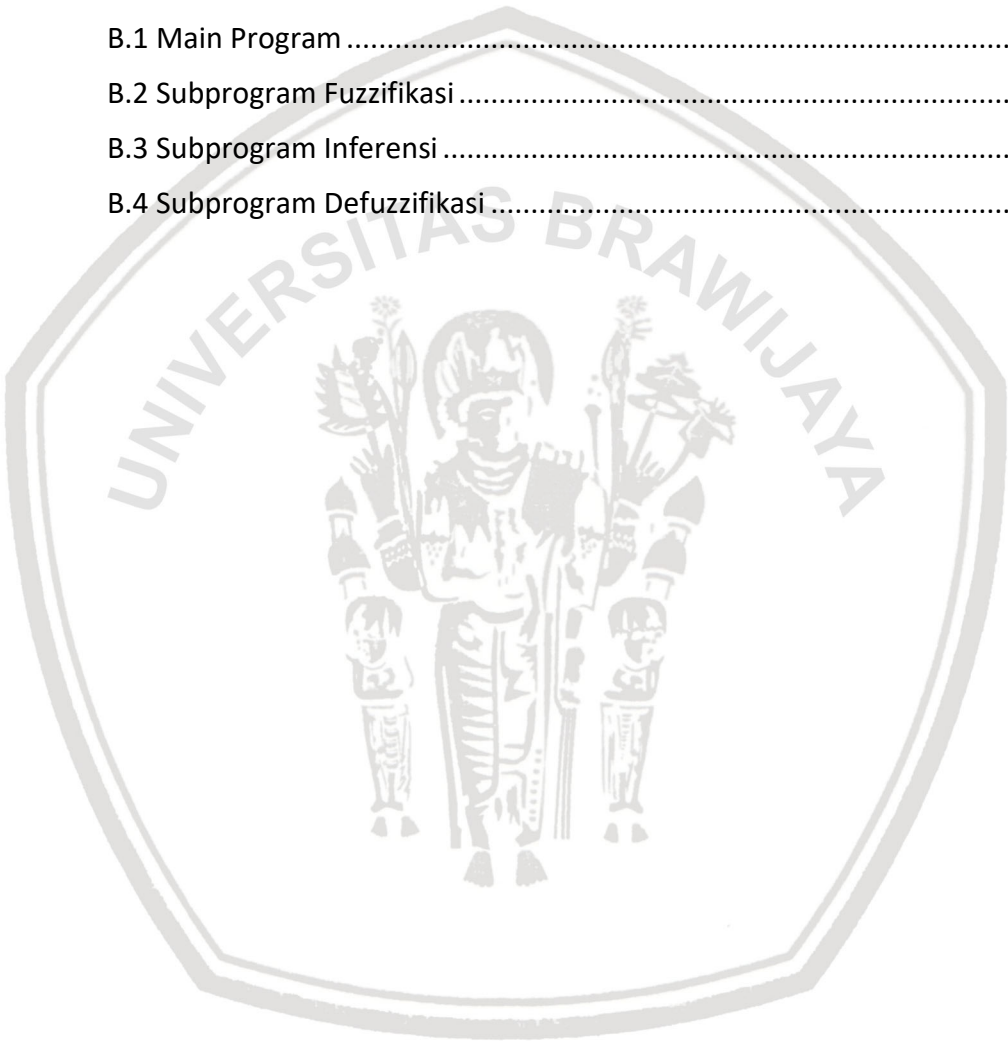
LAMPIRAN B KODE PROGRAM ..... 71

    B.1 Main Program ..... 71

    B.2 Subprogram Fuzzifikasi ..... 73

    B.3 Subprogram Inferensi ..... 74

    B.4 Subprogram Defuzzifikasi ..... 74



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Perkembangan budidaya tanaman jamur tiram pada bidang pertanian banyak dilakukan, seiring dengan populernya produksi jamur tiram sebagai bahan makanan alternatif yang setara dengan daging dan ikan yang mempunyai gizi tinggi. Budidaya tanaman jamur tiram dilakukan dengan cara pembenihan yang dilakukan di bawah kondisi lingkungan yang tertutup. Dengan tempat yang tertutup para peneliti mengalami kesulitan dalam menjaga dan memantau keadaan tempat. Diperoleh kesimpulan bahwa suhu dan kelembapan merupakan faktor terpenting dalam budidaya jamur. Suhu yang dibutuhkan untuk pertumbuhan *Miselim* antara 28-30°C dan kelembapan 70-85%, sedangkan untuk pembentukan tubuh buah (*fruiting body*) 22-26°C dan kelembapan 80-90%. Dan Kubung jamur dianjurkan dibangun pada tempat-tempat teduh dan tidak terkena pancaran sinar matahari secara langsung (Istuti, 2009).

Penelitian suhu dan kelembapan pada budidaya tanaman jamur jika diperoleh derajat kelembapan jamur baik terlalu tinggi maupun rendah, maka kemungkinan besar jamur tidak akan tumbuh, cara yang paling mudah adalah dengan melakukan penyiraman (Suriawiria, 1997). Kebutuhan untuk mengukur parameter fisik memainkan peran penting dalam *sains* dan teknologi, pemanfaatan jaringan sensor dalam penulisan ini dengan membuat komponen elektronik yang dapat menerjemahkan parameter fisik menjadi sinyal listrik.

Air yang sangat dibutuhkan dalam pertumbuhan dan perkembangan tanaman sebanyak 85 hingga 90% dari bobot segar sel-sel dan jaringan tanaman tinggi (Orcott, 1987). Penyiraman dalam implementasi sistem dilakukan dengan menggunakan metode logika *fuzzy* yang dapat dimanfaatkan untuk mengubah metode pengelolaan penyiraman yang ditemukan pada penelitian sebelumnya yang melakukan kendali penyiraman dengan menentukan waktu penyiraman.

*Internet of Things* (IoT) memberi peran penting dalam menerapkan sistem pertanian cerdas. Pertanian cerdas adalah konsep yang muncul karena sensor IoT mampu memberikan informasi tentang bidang pertanian. Pemanfaatan teknologi yang berkembang yaitu IoT dan pertanian cerdas dengan menggunakan otomasi. Pemantauan faktor lingkungan merupakan faktor utama untuk meningkatkan hasil tanaman yang efisien (Prathibha, et al., 2017). Penggunaan sistem pemantau dengan jaringan sensor telah banyak ditemui di berbagai bidang, seperti yang sudah diterapkan pada *smart green house*. Sistem pada *smart green house* memungkinkan penggunaanya untuk mengoperasikan secara jarak jauh dan memantau berbagai aspek kondisi pertanian yang dapat memberikan dasar bagi dukungan keputusan untuk melakukan pemantauan secara *real time* terhadap data informasi yang diperoleh (Zhao & Liu, 2012).

Perkembangan aplikasi kontrol jarak jauh yang dapat dilakukan di mana saja dan kapan saja dan tidak menyita banyak waktu, sebagai contoh pemantauan jarak jauh melalui *handphone* dengan metode *Short Message Service* (SMS). Blynk



adalah platform dengan aplikasi *iOS* maupun *Android* yang digunakan untuk mengendalikan mikrokontroler melalui internet. Kemajuan dalam modul komunikasi dan protokol sangat meningkat, penerapan sistem pemantauan jarak jauh dengan tambahan perangkat aplikasi Blynk. Aplikasi Blynk sudah dapat terhubung dengan *wi-fi*, *ethernet* maupun ESP8266 (Blynk, 2017).

Berdasarkan beberapa penelitian yang telah dilakukan, penelitian untuk membuat suatu sistem pemantauan dan pengendalian penyiraman, yaitu Sistem Pemantauan Menggunakan Blynk dan Pengendalian Penyiraman Tanaman Jamur dengan Metode Logika *Fuzzy*. Hal ini yang mendasari penulis dalam menerapkan metode logika *fuzzy* untuk melakukan kendali penyiraman secara otomatis. Pada penelitian ini, penulis merancang sistem agar dapat melakukan pemantauan suhu dan kelembapan dari jauh dengan menggunakan *smartphone* yang sudah terinstal aplikasi Blynk. Dengan demikian, diharapkan sistem dapat melakukan *pemantauan* dan kendali penyiraman sesuai kondisi yang dibutuhkan.

## 1.2 Rumusan masalah

Berdasarkan uraian latar belakang di atas, berikut ini adalah rumusan masalah penelitian.

1. Bagaimana proses pemanfaatan perangkat sistem pemantauan dan pengendalian penyiraman pada tanaman jamur?
2. Bagaimana implementasi sistem pemantauan dan penyiraman dengan metode logika *fuzzy*?
3. Bagaimana hasil kinerja perangkat sistem pemantauan dan pengendali penyiraman tanaman jamur dengan metode logika *fuzzy* menggunakan Blynk?

## 1.3 Tujuan

1. Mengetahui proses pemanfaatan perangkat sistem pemantauan dan pengendalian penyiraman pada tanaman jamur.
2. Menerapkan perangkat sistem pemantauan dan kendali penyiraman dengan metode logika *fuzzy*.
3. Menguji kinerja perangkat sistem pemantauan menggunakan Blynk dan pengendalian penyiraman tanaman jamur dengan metode logika *fuzzy*.

## 1.4 Manfaat

Dengan adanya perancangan sistem “Sistem pemantauan menggunakan Blynk dan Pengendalian Penyiraman Tanaman Jamur dengan Metode Logika *Fuzzy*”, diharapkan sistem dapat memudahkan peneliti dalam pemantauan dan penyiraman.

Dilihat dari fungsi sistem yang akan diteliti oleh penulis, didapatkan manfaat yaitu sebagai berikut:

1. Menambah ilmu dan wawasan bagi pembaca terkait penelitian sistem pemantauan dan pengendali penyiraman tanaman jamur dengan metode logika *fuzzy* menggunakan Blynk.
2. Suatu kebanggaan bagi diri sendiri ketika mampu membuat sebuah sistem yang dapat membantu meringankan pekerjaan orang.
3. Sebagai bentuk penelitian pemantauan dan penyiraman secara otomatis dengan menggunakan metode logika *fuzzy* yang memungkinkan dapat dikembangkan oleh peneliti lain dengan objek dan metode yang berbeda.

### 1.5 Batasan masalah

1. Pemantauan hanya pada layar LCD dan aplikasi Blynk
2. Implementasi penyiraman menggunakan metode *fuzzy* yang diterapkan pada mikrokontroler NodeMCU.
3. Aplikasi yang digunakan tidak bisa mengontrol penyiraman.

### 1.6 Sistematika pembahasan

Pembahasan dari penelitian dituliskan dengan mengikuti sistematika sebagai berikut.

#### **BAB I PENDAHULUAN**

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan laporan skripsi.

#### **BAB II LANDASAN KEPUSTAKAAN**

Bab ini berisi tentang kajian pustaka terkait dengan penelitian yang telah ada seperti penelitian tentang sistem pemantauan suhu, kelembapan dan pengendali penyiraman *hidroponik* menggunakan Blynk android dan teori-teori yang berhubungan dengan proses pemantauan.

#### **BAB III METODOLOGI PENELITIAN**

Bab ini menjelaskan metode dan langkah-langkah penelitian yang akan digunakan yang berisi alur metode penelitian, studi pustaka, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian dan analisis serta penutup.

#### **BAB IV ANALISIS KEBUTUHAN**

Bab ini berisi analisis kebutuhan seperti gambaran umum sistem, kebutuhan pengguna dan kebutuhan sistem.



**BAB V PERANCANGAN DAN IMPLEMENTASI**

Bab ini berisi tahapan-tahapan dalam melakukan perancangan sistem dan implementasi sistem dengan menerapkan metode logika *fuzzy*.

**BAB VI PENGUJIAN DAN ANALISIS**

Bab ini berisi tentang hasil pengujian akurasi pada sistem yang dibangun dan menganalisis terhadap hasil sistem yang telah direalisasikan.

**BAB VII PENUTUP**

Bab ini berisi kesimpulan yang didapat pada rumusan masalah dan juga saran untuk pengembangan lebih lanjut.

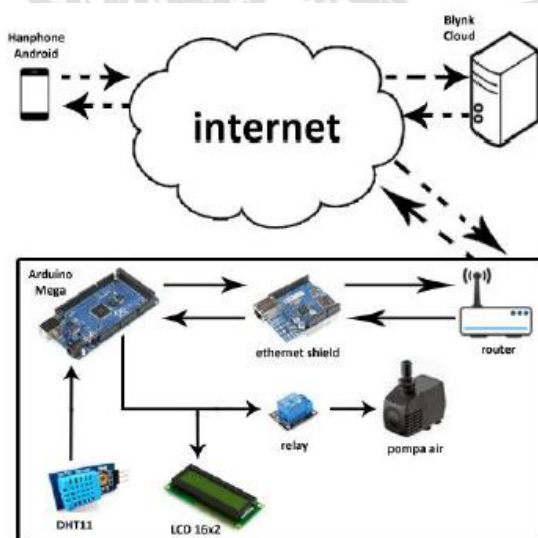


## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

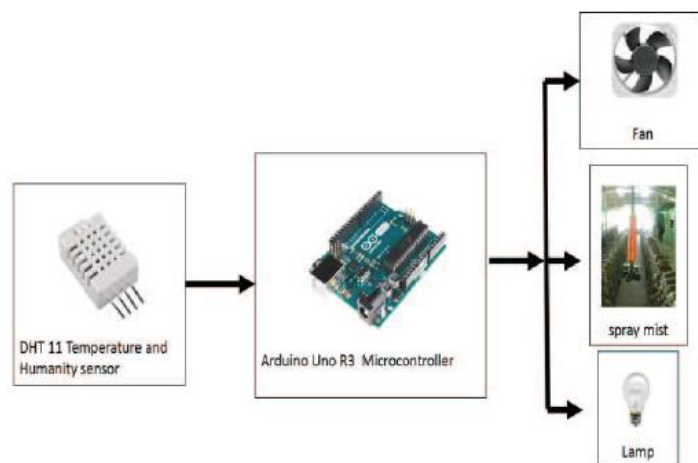
Kajian pustaka membahas mengenai perbandingan penelitian yang pernah dilakukan sebelumnya dengan beberapa objek dan metode yang berbeda sehingga menjadi pendukung dalam proses penelitian yang akan dilakukan. Pada penelitian ini kajian pustaka diambil dari beberapa referensi penelitian sebelumnya. Tujuan dari kajian pustaka yaitu untuk mengkaji hasil penelitian sebelumnya dan dijadikan sebagai dasar acuan dalam membantu pelaksanaan penelitian yang akan dikerjakan.

Penelitian Parayitno (2017) dengan penelitian yang menggunakan perangkat mikrokontroler Arduino mega sebagai sistem akuisisi data dan ditambahkan *ethernal shield* sebagai pengiriman data melalui jaringan internet. Sensor yang digunakan ialah sensor DHT11 untuk mendeteksi suhu dan kelembapan. Aplikasi Blynk digunakan sebagai alat bantu pemantauan dan modul RTC untuk pewaktuan secara *real time*. Pada penelitian tersebut dilakukan penyiraman 3 kali sehari secara otomatis dan penggunaan pemantauan diterapkan pada tanaman hidroponik serta tidak menggunakan metode *fuzzy*.



**Gambar 2.1 Diagram blok sistem yang dikembangkan Prayitno (2017)**

Penelitian lainnya yang dilakukan oleh Kaewwiset & Yodkhad (2017) dengan penelitian yang mengusulkan sistem untuk melakukan pemantauan dan penyiraman pada pembibitan jamur. Peneliti menggunakan Arduino Uno sebagai mikrokontroler dan akuisisi data. Sensor yang digunakan yaitu sensor DHT11 dan terdapat 3 *output* yaitu kipas, penyiraman dan lampu. Pada penelitian ini metode *fuzzy* digunakan untuk melakukan *output* sudah bagus akan tetapi *output* yang diterapkan untuk tanaman hidroponik. Pada penelitian yang akan dilakukan menambahkan aplikasi sebagai kontrol jarak jauh dengan menggunakan koneksi internet.



**Gambar 2.2 Diagram blok sistem yang dikembangkan Kaewwiset & Yodkhad (2017)**

Pada Tabel 2.1 akan menjelaskan persamaan dan perbedaan penelitian-penelitian sebelumnya dengan penelitian yang akan dilakukan.

**Tabel 2.1 Kajian pustaka**

No	Judul	Persamaan	Perbedaan	
			Penelitian Sebelumnya	Rencana Penelitian
1	Sistem <i>Monitoring</i> Suhu, Kelembapan, dan Pengendali Penyiraman Tanaman Hidroponik Menggunakan Blynk Android (Prayitno, 2017)	Sensor DHT11 dan menggunakan Blynk	Pemantauan tanaman hidroponik	Pemantauan tanaman jamur dan menggunakan metode logika <i>fuzzy</i>
2	<i>Automatic temperature and humidity control system by using fuzzy logic alghorithm for mushroom nursery</i> (Kaewwiset & Yodkhad, 2017)	Sensor DHT11, logika <i>fuzzy</i>	Klasifikasi metode <i>neuro fuzzy</i>	Menambahkan aplikasi Blynk dan menggunakan metode logika <i>fuzzy</i> model Takagi Sugeno

## 2.2 Dasar Teori

Dasar teori membahas definisi teori yang berhubungan dengan penelitian yang akan dikerjakan dengan menjabarkan secara sistematis mengenai landasan teori yang digunakan sebagai penguat pemahaman dari kebutuhan sistem yang akan digunakan peneliti, dengan maksud menjelaskan pemahaman secara dasar.



### 2.2.1 Jamur Tiram

Jamur tiram atau dalam bahasa latin *Pleurotus ostreatus* merupakan jenis jamur pangan dengan mempunyai ciri umum tubuh buah berwarna putih hingga krem dan tudungnya berbentuk setengah melingkar mirip cangkang tiram dengan bagian tengah sedikit cekung (Parlindungan, 2000). Kandungan gizi jamur tiram lebih tinggi dibanding dengan jenis jamur kayu lainnya. Jamur tiram memiliki kandungan protein, lemak, fosfor, besi, thiamin dan riboflavin lebih tinggi dibanding dengan jenis jamur lainnya. Jamur tiram mengandung 18 jenis asam amino yang dibutuhkan oleh tubuh manusia dan tidak mengandung kolesterol. Jenis jamur yang telah dibudidaya dan populer sebagai makanan dan sayuran serta banyak diperdagangkan di pasar antara lain jamur tiram. Terdapat beberapa jenis jamur tiram, yaitu tiram putih, tiram abu-abu, sayur caju, tiram coklat (*abalon*) dan jamur tiram merah (Cahaya, et al., 1997). Gambar 2.3 menampilkan bentuk fisik jamur tiram.



**Gambar 2.3 Bentuk fisik jamur tiram**

Sumber: (Ubud Basket, 2018)

Pada umumnya jamur tiram berkembangbiak dengan dua cara dalam siklus hidupnya, yakni secara aseksual maupun seksual (Gunawan & Wydia, 2009). Reproduksi *aseksual* pada jamur terjadi pada jalur *spora* yang terbentuk secara *endogen* yang terletak pada kantung *spora*, *spora aseksual* disebut *konidiospora* karena terbentuk dalam *konidium*. Sedangkan reproduksi secara seksual terjadi karena menyatunya dua jenis *hifa* antara *gamet* jantan dan betina membentuk sel (*zigot*) dari hasil bersatunya dua sel kelamin yang akan tumbuh menjadi *primodia* dewasa.

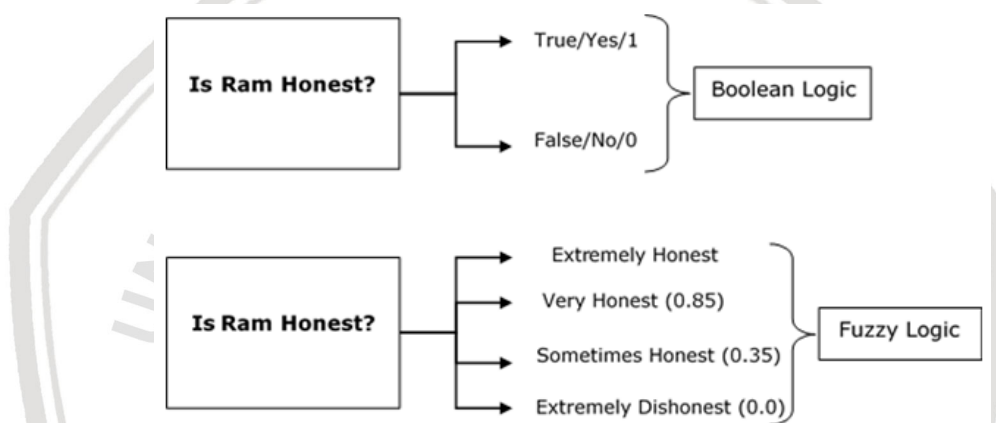
Menurut (Roger, 2006), mula-mula *basidiospora* berkembang membentuk *miselium*, terus bertumbuh hingga *hifa* pada *miselium* saling kompatibel sehingga terjadi *plasmogami* membentuk *hifa dikaryotik*. Setelah itu kondisi lingkungan memungkinkan suhu antara 10-20°C, kelembapan 85-90%, cahaya mencukupi, dan oksigen  $CO_2 < 1000$  ppm maka tubuh buah akan terbentuk (Widiastuti & Panji, 2008).

Pada budidaya jamur tiram suhu dan kelembapan udara memiliki peran yang penting untuk mendapatkan pertumbuhan badan buah yang optimal. Suhu yang

dibutuhkan untuk pertumbuhan *miselium* 20-30°C dan kelembapan 80-85%, suhu untuk pembentukan tubuh buah (*fruiting body*) lebih rendah atau sama dengan 22-26°C. Kubung jamur dianjurkan dibangun pada tempat-tempat yang teduh dan tidak terkena pancaran sinar matahari secara langsung (Nurbana, 2015).

### 2.2.2 Logika Fuzzy

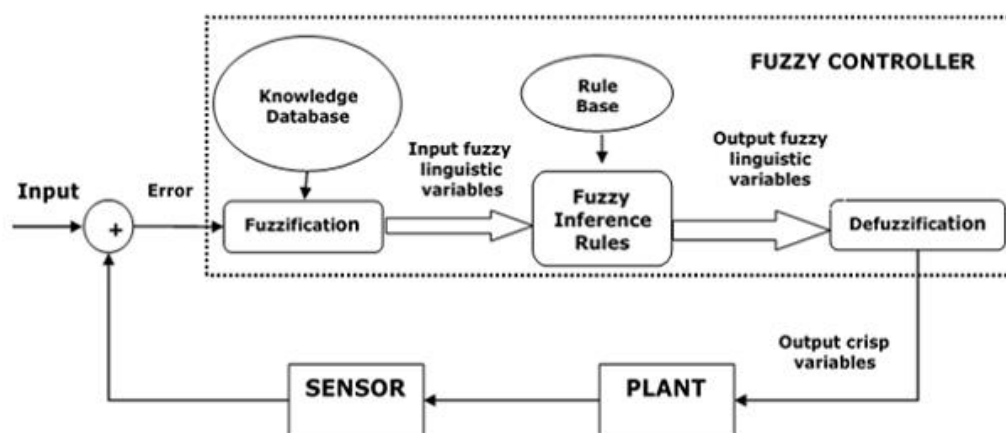
Logika *fuzzy* diterapkan dengan sukses besar dalam berbagai aplikasi kontrol. Hampir semua produk konsumen memiliki kontrol *fuzzy*. Beberapa contoh termasuk mengontrol suhu ruangan dengan bantuan AC, sistem anti-pengereman yang digunakan dalam kendaraan, kontrol lampu lalu lintas, mesin cuci, sistem ekonomi besar, dll. Logika *fuzzy* menyerupai metodologi yang bertujuan sebagai pengambilan keputusan. Nilai keanggotaan pada logika *fuzzy* yaitu antara 0 dan 1, hitam dan putih, ya atau tidak (Zadeh, 1960). Adapun logika *fuzzy* dapat dilihat seperti pada Gambar 2.4.



**Gambar 2.4 Logika fuzzy**

Sumber: (Tutorials Point, 2018)

Model ini diusulkan oleh Takagi Sugeno pada 1985, dengan menggunakan format aturan jika  $x$  adalah  $A$  dan  $Y$  adalah  $B$  THEN  $Z = F(x,y)$ , di mana  $f$  bisa sembarang fungsi dari variabel-variabel *input* yang nilainya berada dalam *interval variabel output*. Biasanya, fungsi ini dibatasi dengan menyatakan  $f$  sebagai kombinasi linier dari variabel-variabel *input*  $f(X_1, \dots, X_n) = W_0 + W_1 X_1 + \dots + W_n X_n$  dimana  $W_0, \dots, W_n$  adalah konstanta berupa bilangan real yang merupakan bagian dari pernyataan aturan *fuzzy*. Fungsi keanggotaan tersebut adalah *singleton* yakni memiliki fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai *crisp* tunggal dan 0 pada semua nilai *crisp* yang lain. Adapun arsitektur kontrol logika *fuzzy* dapat dilihat pada Gambar 2.5.



**Gambar 2.5 Arsitektur logika fuzzy**

Sumber: (Tutorials Point, 2018)

Gambar 2.5 merupakan asitektur diagram control yang menunjukkan komponen-komponen logika fuzzy. Berikut adalah penjelasan komponen-komponen logika fuzzy yaitu:

1. *Fuzzifier* untuk megkonversi nilai *input* yang ditetapkan ke nilai *fuzzy*.
2. Basis pengetahuan *fuzzy* untuk mendefinisikan keanggotaan variabel *input* ke basis aturan *fuzzy* dan variabel *output* ke dalam bagian kontrol.
3. *Fuzzy rule base* untuk menyimpan mengenai proses operasi domain.
4. *Inferensi* untuk mensimulasikan keputusan manusia dengan melakukan penalaran perkiraan.
5. *Defuzzifier* untuk mengubah nilai *fuzzy* menjadi nilai-nilai yang sudah ditentukan

### 2.2.3 Mikrokontroler

Mikrokontroler adalah komputer mikro yang terdapat pada sebuah *chip* tunggal yang digunakan sebagai pengontrol rangkaian elektronik dan juga dapat menyimpan program. Pada umumnya setiap mikrokontroler terdiri dari memori, *Central Processing Unit* (CPU), I/O dan di dalamnya sudah terintegrasi *Analog to Digital Converter* (ADC). Pada Gambar 2.6 merupakan beberapa jenis mikrokontroler.

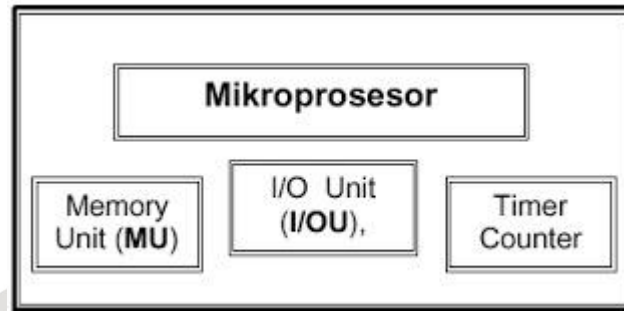


**Gambar 2.6 IC mikrokontroler**

Sumber : (Jogja Robotika, 2018)



Fungsi dari mikrokontroler yaitu sebagai *counter*, *decoder* dan *encoder*, *flip-flop*, pembangkit orisinal, pembangkit osilasi, *timer*/pewaktu, Sebagai ADC dan masih banyak lagi. Dengan kata lain mikrokontroler dapat dikatakan sebagai suatu alat elektronik digital yang mempunyai I/O dan kontrol dengan program yang bisa ditulis dan dihapus dengan cara tertentu. Adapun Gambar 2.7 menampilkan diagram blok pada mikrokontroler.



**Gambar 2.7 Blok diagram mikrokontroler**

Sumber: (Immersa Lab, 2014)

Dengan adanya mikrokontroler dapat mengurangi ukuran, biaya dan konsumsi tenaga dibandingkan menggunakan mikroprosesor memori dan alat I/O yang terpisah, kehadiran mikrokontroler juga dapat membuat proses kontrol elektrik menjadi lebih ekonomis, seperti:

1. Sistem elektronik akan menjadi ringkas, karena memori, I/O dan prosesor sudah terintegrasi menjadi satu, dan dapat bekerja secara inovatif sesuai yang dibutuhkan sistem.
2. Tersedianya fasilitas untuk melakukan pengembangan memori dan I/O yang dapat disesuaikan dengan kebutuhan sistem.
3. Kontrol pada mikrokontroler menggunakan pemrograman *assembly* sehingga pengoperasian sistem lebih mudah dikerjakan sesuai dengan kebutuhan sistem.

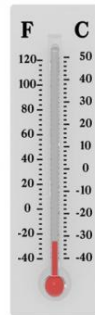
#### 2.2.4 Sensor

Sensor adalah suatu jenis transduser yang dapat digunakan untuk mengubah besaran lingkungan fisik maupun kimia menjadi arus listrik. Sensor sering kali digunakan sebagai pendeteksi saat melakukan pengendalian maupun pengukuran.

##### 1. Suhu

Suhu sangat berpengaruh terhadap ekosistem, karena suhu memiliki fungsi yang diperlukan bagi makhluk hidup. Makhluk hidup memiliki suhu optimal untuk kelangsungan hidup. Hal ini disebabkan karena perubahan materi yang menyangkut struktur dalam molekul suatu zat yang ada di dalam tubuh terhadap pengaruh kualitas suhu pada lingkungan. Suhu pada umumnya menunjukkan derajat panas benda, jika semakin tinggi suhu

suatu benda maka semakin panas juga benda tersebut. pada Gambar 2.8 menampilkan pengukuran suhu menggunakan termometer.



**Gambar 2.8 Pengukuran suhu pada termometer**

Sumber: (College Loan Consolidation, 2016)

## 2. Kelembapan

Kelembapan dibutuhkan karena dapat memengaruhi kecepatan uap air dari tubuh makhluk hidup untuk bertahan hidup terhadap lingkungan yang kering. Kelembapan juga memengaruhi perkembangan dan pertumbuhan tanaman agar batangnya tidak cepat kering karena terjadinya proses penguapan. Pada setiap tanaman memiliki kebutuhan kelembapan yang berbeda-beda tergantung jenis tanaman, ada yang membutuhkan kelembapan tinggi dan ada juga yang rendah. Kelembapan udara sangat berpengaruh terhadap penguapan pada permukaan tanah dan permukaan daun. Kelembapan udara berpengaruh terhadap kecepatan penguapan, jika kelembapan rendah maka dapat menyebabkan kecepatan penguapan, penyerapan air dan zat-zat mineral meningkat. Sedangkan jika kelembapan tinggi maka kecepatan penguapan rendah dan zat-zat nutrisi juga berkurang. Pada Gambar 2.9 menampilkan pengukuran kelembapan menggunakan higrometer.

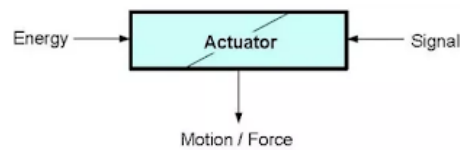


**Gambar 2.9 Pengukur kelembapan pada higrometer**

Sumber : (Alat Labor, 2018)

### 2.2.5 Aktuator

Aktuator adalah sebuah peralatan mekanis yang digunakan untuk mengontrol sebuah sistem. pada umumnya aktuator digerakkan dengan motor listrik yang dikendalikan melalui media pengontrol otomatis yang sudah terprogram di antaranya mikrokontroler. Pada Gambar 2.10 menampilkan fungsi aktuator.



**Gambar 2.10 Aktuator**

Sumber: (Stand Out Publishing, 2017)

Aktuator memiliki fungsi untuk menghasilkan gerakan rotasi maupun translasi, seperti otomatisasi kendali pintu, dengan mengubah sinyal listrik menjadi besaran mekanik. Aktuator memiliki beberapa jenis sebagai tenaga penggerak antara lain:

1. Aktuator tenaga elektrik

Pada umumnya menggunakan solenoid, motor arus searah (DC). Karena sifatnya mudah diatur sesuai dengan pemutaran kecil sampai sedang. Aktuator elektrik mempunyai fungsi yang dapat mengubah sinyal elektrik menjadi gerakan mekanik.

2. Aktuator tenaga *hidrolik*

Pada umumnya memiliki susunan pemutaran yang besar sehingga sulit untuk diselesaikan. Aktuator *hidrolik* membutuhkan tekanan aliran *fluida* agar menjadi gerakan mekanik.

3. Aktuator tenaga *pneumatic*

Aktuator ini memiliki konstruksi yang sulit untuk dikendalikan contohnya seperti gerakan pada piston, karena membutuhkan tekanan udara agar mendapatkan gerakan mekanik.

### 2.2.6 Blynk

Blynk adalah sebuah platform aplikasi *Operating System (OS) Mobile* seperti iOS dan Android yang bertujuan untuk memudahkan dalam pengendali modul seperti Arduino, Raspberry Pi, ESP8266, WEMOS *D1* dan module jenis lain melalui internet. Platform ini merupakan dasbor digital sebagai tempat kreativitas untuk membuat antarmuka grafis untuk proyek yang akan diimplementasikan hanya dengan metode *drag* dan *drop*. Dari aplikasi ini dapat mengontrol dari jarak jauh, kapanpun dan dimanapun kita berada. Aplikasi harus terhubung melalui internet dengan koneksi yang stabil (Blynk, 2017). Gambar 2.11 menampilkan antarmuka aplikasi Blynk.



**Gambar 2.11 Antarmuka aplikasi Blynk**

Sumber: (Blynk, 2017)



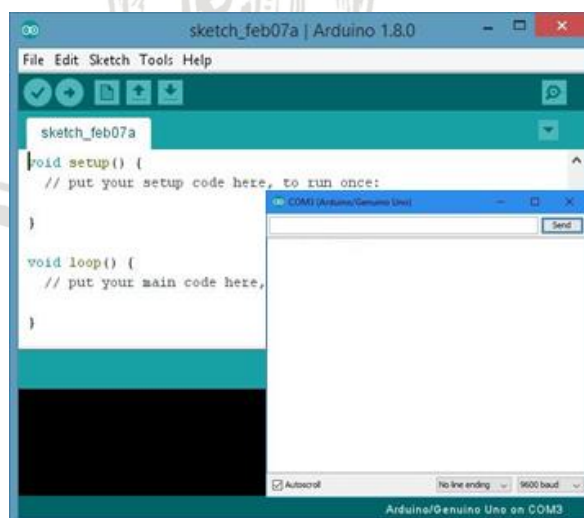
Adapun fitur-fitur yang ada pada aplikasi Blynk yaitu sebagai berikut:

1. API & UI serupa untuk semua perangkat keras & perangkat yang didukung
2. Dapat terhubung ke cloud menggunakan:
  - Wi-Fi
  - Bluetooth
  - BLE Ethernet USB (serial)
  - GSM, dan lain-lain
3. Manipulasi pin langsung tanpa penulisan kode
4. Mudah untuk mengintegrasikan dan menambahkan fungsi baru menggunakan pin *virtual*
5. Riwayat *pemantauan* data melalui *widget graph history*
6. Komunikasi *device to device* menggunakan *bridge widget*, serta fitur-fitur baru yang terus ditambahkan

### 2.2.7 Arduino IDE

Bahasa pemrograman C adalah sebuah bahasa pemrograman komputer yang bisa digunakan untuk membuat berbagai aplikasi (*general purpose programming language*), bisa disebut sebagai bahasa pemrograman tingkat menengah dan juga termasuk dalam bahasa tingkat tinggi. Mulai dari sistem operasi seperti Windows atau Linux, *software* pengolah gambar (*image processing*), hingga *compiler* untuk bahasa pemrograman, di mana pemrograman C banyak digunakan salah satunya adalah Arduino IDE.

Dengan menggunakan pemrograman berbasis bahasa C dapat digunakan untuk membuat kode program yang digunakan di Arduino karena bersifat *open source* dengan menggunakan *toolchains compiler* dan juga menyediakan *integrated development environment* (IDE) untuk menuliskan *source code* program yang disebut dengan istilah *sketch*. Seperti yang diperlihatkan pada Gambar 2.12 menampilkan *sketch* yang terdapat pada Arduino IDE.



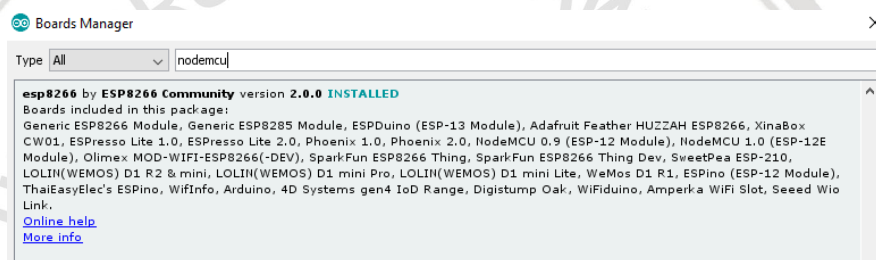
Gambar 2.12 Sketch Arduino IDE

Program yang dituliskan melalui Arduino IDE disebut sebagai sketsa. Pada Tabel 2.2 menjelaskan sebuah sketsa Arduino yang mempunyai fungsi sebagai berikut.

**Tabel 2.2 Sketsa Arduino IDE**

Fungsi	Keterangan
<i>Setup()</i>	Dipanggil sekali setelah <i>power-up</i> atau <i>reset</i> , biasanya digunakan untuk menginisialisasi variabel, mode pin, atau proses lain yang diperlukan dalam sketsa.
<i>Loop()</i>	Dipanggil setelah fungsi <i>setup()</i> selesai dijalankan, dieksekusi berulang-ulang sampai <i>power-off</i> atau <i>reset</i> .

Fitur *serial monitor* pada Arduino IDE digunakan dalam proses kirim dan terima data antara PC dan mikrokontroler melalui komunikasi serial. Pengembang bisa memanfaatkan *serial monitor* untuk melakukan *debugging* terhadap sketsa yang sedang dijalankan oleh mikrokontroler. Fitur penting lainnya dari Arduino IDE adalah *Board Manager*, seperti yang diperlihatkan pada Gambar 2.13.



**Gambar 2.13 Board manager Arduino IDE**

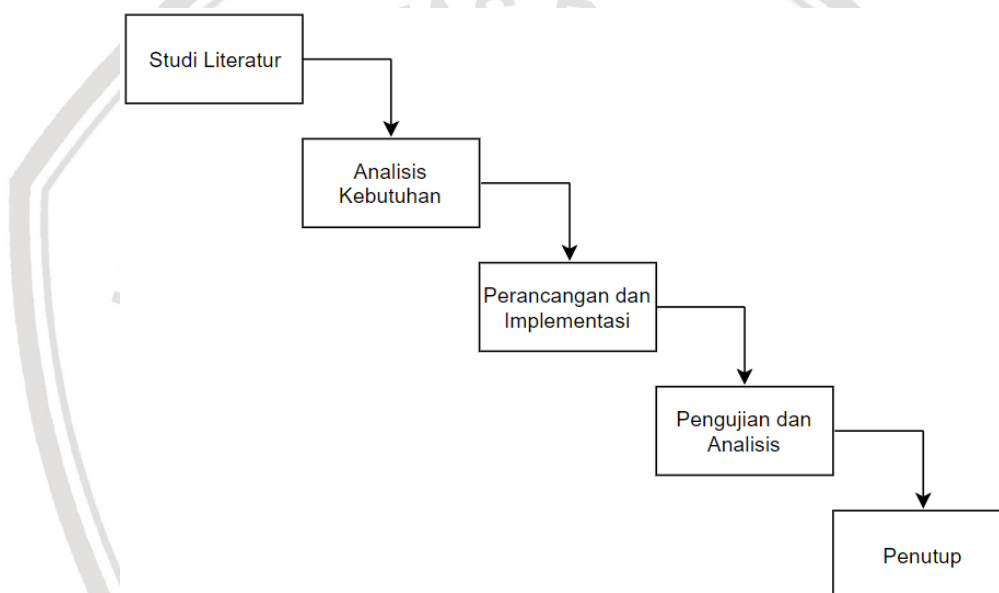
*Board Manager* digunakan untuk menginstal Arduino Core, yaitu sekumpulan komponen perangkat lunak yang dibutuhkan untuk mengkompilasi sketsa ke bahasa mesin mikrokontroler tujuan. Pengembangan ESP8266 dengan Arduino dimungkinkan berkat dukungan yang diperoleh melalui *Arduino core for ESP8266* (ESP8266 Community Forum, 2018). *Arduino core for ESP8266* menyediakan berbagai pustaka untuk berkomunikasi melalui jaringan *wi-fi* menggunakan *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP), menyiapkan *server Hypertext Transfer Protocol* (HTTP), *Domain Name System* (DNS), melakukan pembaruan *Over The Air* (OTA), menggunakan *filesystem* dalam *flash memory*, bekerja dengan kartu *Secure Digital* (SD), *servo*, *periferal* Sistem Pengendalian Intern (SPI) dan *Inter Integrated Circuit* (I2C).

## BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan langkah-langkah terkait alur metode penelitian, studi literatur, analisis kebutuhan, perancangan dan implementasi, pengujian dan analisis serta penutup. Dengan diterapkannya langkah-langkah metodologi penelitian diharapkan sistem dapat dikerjakan secara tertata dan rinci.

### 3.1 Alur Metode Penelitian

Alur metode penelitian yang akan dilakukan bersifat implementatif dengan sub tipe pengembangan atau *development*. Penelitian dimulai dari beberapa tahap yaitu studi pustaka, analisis kebutuhan, perancangan, implementasi, pengujian, analisis sistem kemudian dilanjutkan dengan mengambil kesimpulan. Pada Gambar 3.1 merupakan diagram alir yang berisi langkah-langkah yang dilakukan dalam penelitian.



Gambar 3.1 Diagram alur metodologi penelitian

### 3.2 Studi Literatur

Studi literatur dilakukan untuk memudahkan langkah-langkah dalam mencari jurnal yang bersumber dari penelitian sebelumnya yang dapat mendukung metode penelitian, analisis kebutuhan, perancangan dan implementasi, serta proses pengujian dan analisis sistem yang akan dikerjakan. Berikut teori-teori yang dijadikan sebagai acuan dalam mendukung penelitian yaitu:

#### 1. Teori Terkait Jamur Tiram

Melakukan kajian mengenai bagaimana cara budidaya jamur tiram agar tumbuh dengan baik dan siap di panen, suhu dan kelembapan sangat di butuhkan untuk melihat kadar udara yang ada di dalam kubung jamur



kemudian penyiraman digunakan apabila dibutuhkan seperti saat suhu panas dan kelembapan kering.

2. Teori terkait logika *fuzzy*

Melakukan kajian tentang menentukan nilai ketidakjelasan atau bermakna ganda, seperti mengambil keputusan nilai keanggotaan antara ya atau tidak, 0 dan 1.

3. Teori terkait mikrokontroler

Melakukan kajian mengenai pengertian, fungsi dan manfaat mikrokontroler sebagai CPU, I/O dan memori penyimpan data.

4. Teori terkait Sensor

Melakukan kajian tentang sensor yang dapat mengambil data suhu dan kelembapan secara bersamaan dengan menggunakan 1 sensor.

5. Teori terkait aktuator

Melakukan kajian tentang aktuator, terutama terkait bagaimana untuk menghasilkan kendali gerakan rotasi maupun translasi gerakan dari sinyal atau energi yang di dapat.

6. Teori terkait Blynk

Melakukan kajian tentang spesifikasi dan konfigurasi menu *widget* pada Blynk agar dapat terhubung dengan mikrokontroler.

7. Teori terkait Arduino IDE

Melakukan kajian tentang penggunaan kode pemrograman Bahasa C yang akan ditulis pada *software* Arduino IDE.

### 3.3 Analisis Kebutuhan

Analisis kebutuhan diperlukan untuk mengetahui apa saja komponen-komponen yang akan dibutuhkan oleh sistem yang akan digunakan sebagai penunjang dalam membuat penelitian. Adapun kebutuhan yang diperlukan yaitu gambaran umum sistem, kebutuhan pengguna, kebutuhan sistem, kebutuhan fungsional dan kebutuhan *non-fungsional*. Gambaran umum sistem berisi rincian yang harus dimiliki agar dapat melakukan perancangan, kebutuhan pengguna berisi fitur yang digunakan untuk memudahkan dalam pemantauan.

Kebutuhan sistem dibagi menjadi 2 jenis yaitu kebutuhan perangkat keras dan kebutuhan perangkat lunak. Kebutuhan perangkat keras membahas mengenai spesifikasi perangkat keras yang digunakan. Pada kebutuhan perangkat lunak membahas perangkat yang dibutuhkan untuk mendukung performa sistem. Pada analisis kebutuhan fungsional berisi proses-proses yang nantinya akan dilakukan oleh sebuah sistem. Sedangkan kebutuhan *non-fungsional* mengacu pada batasan layanan atau properti yang dimiliki oleh sistem.

### 3.4 Perancangan dan Implementasi

Perancangan dan implementasi menjelaskan pengerjaan dari sisi perangkat keras dan perangkat lunak agar menjadi satu kesatuan sistem yang akan digunakan sebelum melakukan pengujian.

#### 3.4.1 Perancangan

Perancangan sistem berisi tahapan-tahapan beberapa komponen untuk menggambarkan perencanaan bentuk sistem menyangkut konfigurasi dari komponen perangkat keras dan perangkat lunak dari suatu sistem. Pada perancangan perangkat keras terdapat beberapa subsistem yang akan dirancang agar membentuk komponen secara keseluruhan, sedangkan perancangan perangkat lunak terdapat perancangan program, metode *fuzzy* dan perancangan aplikasi Blynk.

#### 3.4.2 Implementasi

Implementasi adalah tahap pelaksanaan rancangan yang telah disusun secara rinci dan disesuaikan dengan kebutuhan sistem dan perancangan sistem. Implementasi sistem dilakukan dengan menaruh sistem pada kubung jamur untuk pemantauan suhu dan kelembapan, kemudian melakukan penyiraman dengan *rule fuzzy* apabila dibutuhkan untuk penyiraman.

### 3.5 Pengujian dan Analisis

Pengujian dan analisis dilakukan untuk mengetahui kinerja keseluruhan sistem yang sudah dirancang. Tujuannya untuk mengetahui hasil perancangan dan implementasi sistem yang sudah dibuat yang akan dijelaskan pada pengujian dan analisis. Dalam pengujian dilakukan 3 jenis pengujian yaitu pengujian sensor, pengujian pemantauan dan pengujian penyiraman menggunakan metode logika *fuzzy*. Pada pengujian sensor menganalisis perbandingan antara sensor DHT11 dengan termometer yang sama-sama mendeteksi suhu namun, pada termometer tidak bisa mendeteksi kelembapan. Pada pengujian pemantauan menganalisis cara memantau kondisi kubung jamur dengan melihat suhu dan kelembapan, perbandingan nya ketika menggunakan Blynk sistem dapat di pemantau dari jarak jauh. Pada pengujian penyiraman menganalisis perbandingan dengan mengukur banyaknya air yang digunakan dan menghitung lama waktu penyiraman.

### 3.6 Penutup

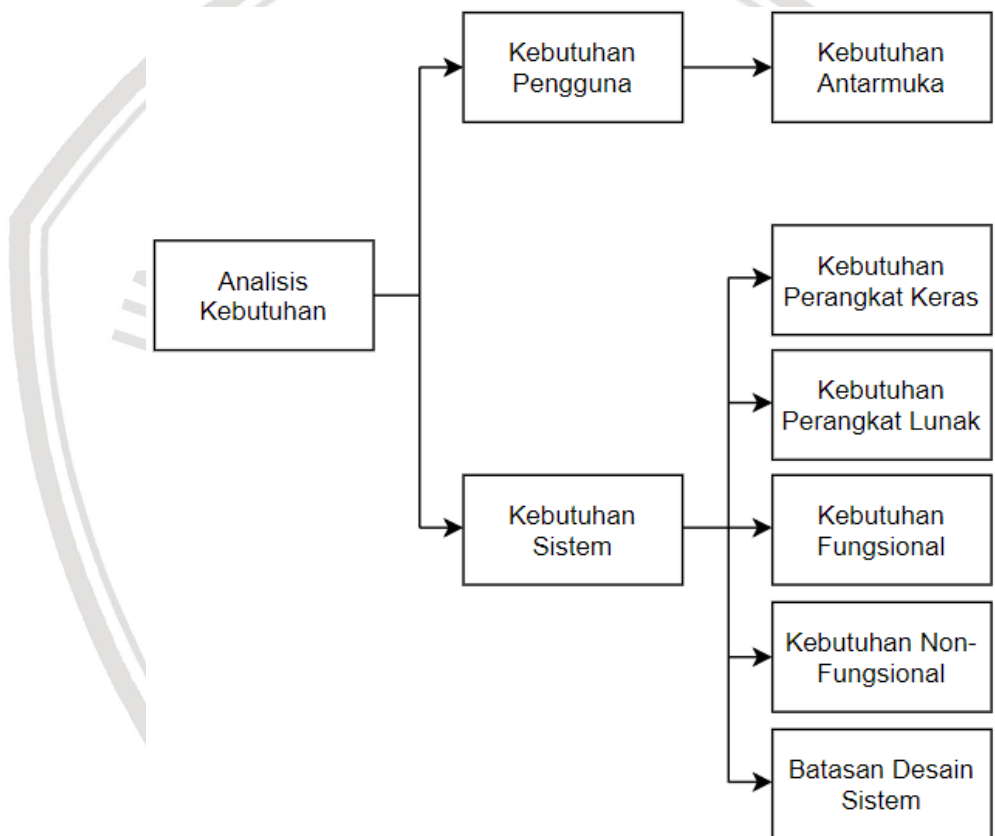
Pada penutup berisi bagian kesimpulan yang diambil dari rumusan masalah yang diperoleh dari hasil yang sudah dilakukan yaitu dengan melihat hasil sistem pada saat melakukan proses perancangan dan implementasi serta pengujian dan analisis apakah sudah sesuai dengan rumusan masalah atau belum. Selain itu, pada akhir penulisan berisi saran yang bertujuan sebagai pertimbangan penelitian selanjutnya agar sistem yang dikembangkan menjadi lebih baik lagi.

## BAB 4 ANALISIS KEBUTUHAN

Bab ini menjelaskan terkait gambaran umum sistem, kebutuhan pengguna dan kebutuhan sistem yang harus terpenuhi agar dapat melakukan tahap selanjutnya yaitu perancangan dan implementasi. Dengan adanya analisis kebutuhan diharapkan sistem dapat menggunakan komponen-komponen yang sudah sesuai agar berjalan dengan baik dan sesuai dengan harapan.

### 4.1 Gambaran Umum Sistem

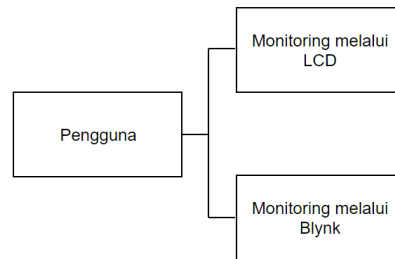
Gambaran umum menjelaskan rincian mengenai analisis kebutuhan yang harus dimiliki sistem agar dapat melakukan perancangan dan implementasi agar sistem dapat bekerja dengan baik dan benar. Pada Gambar 4.1 menampilkan kebutuhan-kebutuhan yang akan dijelaskan pada analisis kebutuhan.



Gambar 4.1 Diagram analisis kebutuhan

### 4.2 Kebutuhan Pengguna

Kebutuhan pengguna untuk memudahkan fungsi dan fitur sistem dalam melakukan pemantauan yang dapat menampilkan hasil data informasi sensor ke pengguna melalui LCD dan aplikasi Blynk. Kebutuhan pengguna dapat dilihat pada Gambar 4.2.



**Gambar 4.2 Kebutuhan pengguna**

Pengguna dapat melihat hasil informasi data suhu dan kelembapan melalui:

1. Pemantauan melalui LCD

Pemantauan melalui LCD dilakukan dengan melihat hasil data informasi dengan mengunjungi kubung jamur.

2. Pemantauan melalui Blynk

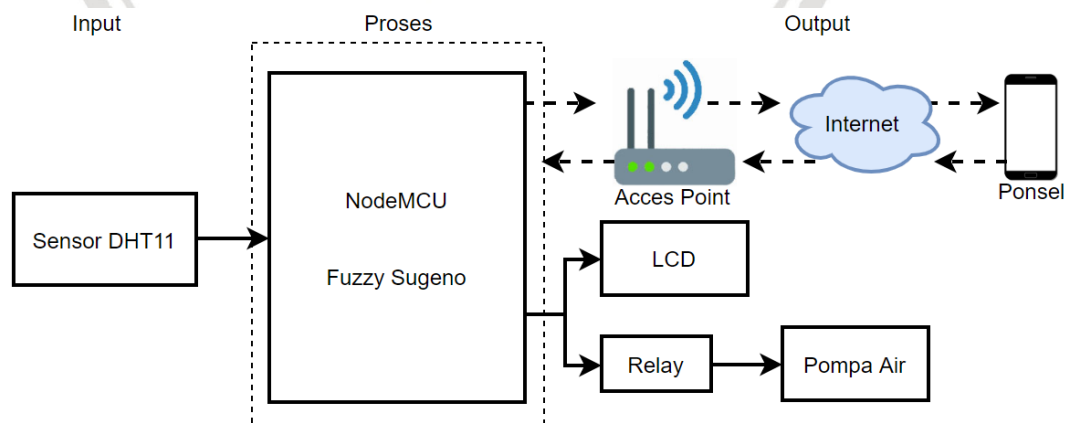
Pemantauan melalui Blynk cukup dilakukan dengan membuka aplikasi Blynk yang sudah terhubung dengan konfigurasi *wi-fi* pada ESP8266 NodeMCU, agar dapat melihat hasil data sensor dari jarak jauh.

#### 4.2.1 Kebutuhan Antarmuka

Pada kebutuhan antarmuka pengguna fungsi sebagai *interface* antara pengguna dan layar LCD maupun aplikasi Blynk. Dari hasil pertemuan dapat melihat hasil informasi data suhu dan kelembapan dengan melihat secara langsung pada kubung jamur melalui LCD dan apabila ingin di pantau dari jauh pengguna bisa melihat melalui aplikasi Blynk.

#### 4.3 Kebutuhan Sistem

Pada bab ini membahas kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan sistem dianalisis sesuai dengan yang dibutuhkan oleh sistem sehingga diharapkan dapat mempermudah dalam melakukan tahap perancangan dan implementasi sistem. pada Gambar 4.3 menampilkan diagram blok sistem yang akan dikerjakan.



**Gambar 4.3 Diagram blok sistem**



Pada Gambar 4.3 Kebutuhan sistem memiliki 3 bagian agar sistem dapat berjalan yaitu *input*, proses dan *output*. Pada bagian *input* berisi sensor DHT11 yang digunakan sebagai alat untuk mengambil data suhu dan kelembapan yang ada di sekitar ruangan. Pada bagian proses terdapat modul mikrokontroler NodeMCU yang digunakan untuk mengakuisisi data yang masuk dari sensor dan juga sebagai kontrol penghubung dengan aktuator untuk meneruskan proses selanjutnya yaitu *output*, pada bagian proses juga terdapat logika *fuzzy* dengan menggunakan model Takagi Sugeno sebagai metode yang diterapkan dalam melakukan pengendali penyiraman. Sedangkan bagian *ouput* berisi internet untuk menghubungkan sistem dengan aplikasi Blynk agar dapat melakukan pemantauan dari jarak jauh melalui perantara *wi-fi* yang sudah ada pada perangkat NodeMCU ESP8266 dengan dihubungkan pada perangkat *smartphone*, fungsi LCD sama seperti Blynk sebagai pemantauan akan tetapi pemantauan pada LCD harus dilakukan dengan melihat hasil sensor pada tempat peletakan sistem. sedangkan relay digunakan sebagai kendali *switch on-off* untuk menggerakan pompa dalam melakukan penyiraman.

#### 4.3.1 Kebutuhan Perangkat Keras

Pada kebutuhan sistem diperlukan perangkat keras untuk menunjang dalam melakukan perancangan dan implementasi. Adapun perangkat keras yang diperlukan seperti NodeMCU, sensor DHT11, relay, pompa air, LCD, LM2596, *smartphone* dan *power bank*. Alasan kebutuhan perangkat keras yang sudah disebut akan dijelaskan secara rinci.

##### 4.3.1.1 NodeMCU

NodeMCU adalah sebuah *platform Internet of Things* yang bersifat *open source* yang di buat oleh *Espressif System* yang terdiri dari perangkat keras berupa *System On Chip* ESP8266, serta *firmware* yang digunakan memakai bahasa pemrograman *scripting Lua* (Nodemcu, 2014). NodeMCU dapat dianalogikan sebagai papan arduino ESP8266. Selain fungsionalitas *wi-fi*, ESP8266 juga mengintegrasikan *mikroprosesor* dan SRAM. Sehingga ESP8266 dapat dihubungkan dengan sensor eksternal dan perangkat lainnya melalui pin *General Purpose Input Output* (GPIO).

NodeMCU yang berbasis ESP8266 dirancang untuk perangkat yang bersifat *mobile*, perangkat *wearable* dan perangkat *Internet of Things* serta perangkat lainnya yang cenderung mengonsumsi daya rendah. Serta berkomunikasi dengan menggunakan ekstensi *Universal Serial Bus* (USB) *to serial* yang biasa digunakan sebagai kabel data pada *smartphone* Android. Papan mikrokontroler ini biasanya diaplikasikan untuk perangkat *smarthome*, *home automation*, *industrial wireless control*, *IP camera*, *mesh network*, *wearable electronic*, *security id tags*, *wi-fi position system beacon*, dll.

NodeMCU telah banyak didukung bahasa pemrograman antara lain adalah *AT Command* dan *Scripting Lua* sebagai bahasa resmi untuk NodeMCU, lalu terdapat dukungan bahasa seperti C++ melalui library pendukung yang perlu dipasang pada suatu kode editor yaitu Platform.IO serta telah memiliki banyak dukungan *library*

yang dapat digunakan pada kode editor Arduino seperti Arduino IDE sehingga untuk membuat program pada papan mikrokontroler ini menjadi lebih mudah dan cepat. Tabel 4.1 memperlihatkan spesifikasi ESP8266.

**Tabel 4.1 Spesifikasi NodeMCU**  
Sumber : (Espressif Systems, 2017)

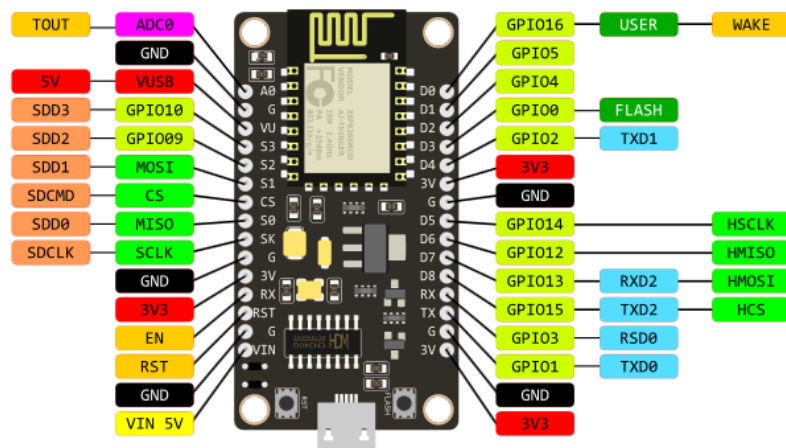
Kategori	Item	Parameter
Wi-Fi	Protokol	802.11 b, g, n, e, i
	Rentang Frekuensi	2,4 – 2,5 GHz
	Keamanan, Enkripsi	WPA/WPA2, WEP, TKIP, AES
	Mode	Station, SoftAP, SoftAP+Station
Hardware	CPU	Tensilica L106 32-bit <i>micro controller</i>
	Antarmuka Periferal	UART, SDIO, SPI, I <sup>2</sup> C, I <sup>2</sup> S, IR <i>remote control</i> , GPIO, ADC, PWM, LED, <i>button</i>
	Tegangan Kerja	2,5 – 3,6V
	Suhu Kerja	-40 – 125°C
Software	Pembaruan <i>Firmware</i>	UART <i>download</i> , OTA
	Protokol Jaringan	IPv4, TCP, UDP, HTTP, FTP

Proyek (NodeMCU Team, 2014) memiliki tujuan untuk menyederhanakan pengembangan ESP8266. Salah satu inisiatif yang dilakukan adalah dengan mengembangkan *development board* yang menggabungkan *chip* ESP8266 pada papan sirkuit standar, seperti yang diperlihatkan pada Gambar 4.4.



**Gambar 4.4 NodeMCU development kit**  
Sumber : (Seed Studio, 2017)

Papan tersebut memiliki *built-in* USB *port* yang sudah terhubung dengan *chip*, sehingga bisa diprogram dengan menghubungkannya ke laptop melalui kabel mikro USB. Selain itu terdapat juga tombol *reset*, antena *wi-fi*, lampu *LED*, dan pin GPIO berukuran standar yang bisa ditancapkan ke *project board*. Pada Gambar 4.5 memperlihatkan skema dari pin-pin NodeMCU.



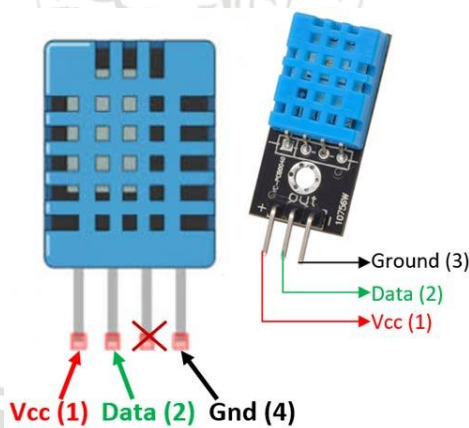
**Gambar 4.5 Skema pin-pin NodeMCU**

Sumber : (Juraj, 2018)

Pada penelitian ini menggunakan modul NodeMCU sebagai masukan data I/O, penyimpan data dan kontrol data. Selain memiliki platform *opensource*, NodeMCU dipilih karena memiliki ukuran board yang kecil yang memiliki *system on chip* ESP8266 yang memiliki kapabilitas akses *wi-fi* dan juga chip komunikasi *USB to serial*. Sehingga memudahkan pengguna untuk melakukan ekstensi data dengan kabel USB yang biasa digunakan untuk mengisi daya pada *smartphone* Android.

#### 4.3.1.2 Sensor DHT11

Sensor DHT11 merupakan sensor yang mengukur 2 parameter lingkungan sekaligus yang berfungsi untuk mendeteksi suhu dan kelembapan. Pada Gambar 4.6 menampilkan bentuk skematik sensor DHT11.



**Gambar 4.6 Skematik sensor DHT11**

Sumber : (Zhu, 2018)

Pada Gambar 4.6 menjelaskan sensor DHT11 yang memiliki sebuah *thermistor* tipe NTC (*Negative Temperature Coefficient*) untuk mengukur suhu sedangkan pada kelembapan bertipe *resistif* dan sebuah mikrokontroler 8-bit yang mengolah kedua sensor tersebut dan hasilnya dikirimkan ke pin *output* dengan format *single-wire bi-directional* (kabel tunggal dua arah). Jadi, meskipun sensor ini terlihat kecil, DHT11 ternyata mampu melakukan fungsi yang cukup kompleks. Setelah kita

mengambil data dan mendapatkan *output* nya, kemudian dimasukkan ke sistem yang kita uji (Ajie, 2016). Pada Tabel 4.2 menunjukkan spesifikasi sensor DHT11.

**Tabel 4.2 Spesifikasi sensor DHT11**

Sumber : (Components101, 2018)

Fungsi	Sensor DHT11
<i>Resolution</i>	16-bit
<i>Voltage</i>	3.5-5.5V
<i>Current</i>	0.3mA ( <i>measuring</i> ) 60uA ( <i>standby</i> )
<i>Output</i>	Serial Data
<i>Range</i>	0°C-50°C & 20-90%
<i>Accuracy</i>	±1°C & ±1%

Pada umumnya terdapat dua macam tipe DHT11 yang sudah berupa modul seperti DHT11 dengan 3 pin dan 4 pin. Adapun fungsinya sama saja, karena pada modul DHT11 yang berkaki 4 ada satu pin yang tidak digunakan yaitu pin *Normally Close* dan tidak menggunakan *breakout* PCB. Pada Tabel 4.3 berikut adalah fungsi konfigurasi dari pin DHT11.

**Tabel 4.3 Konfigurasi pin DHT11**

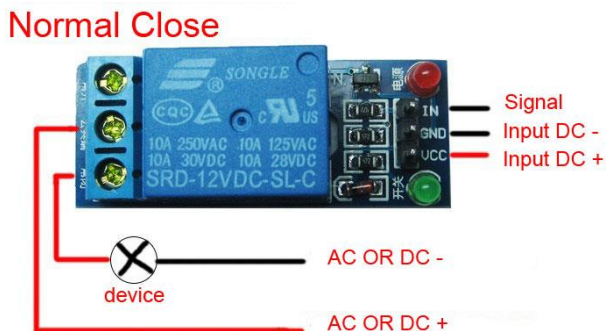
DHT11 Pin	Fungsi
Pin 1	Vcc 3.5 – 5.5v DC
Pin 2	Data/ Serial data ( <i>single bus</i> )
Pin 3	Ground

Sensor DHT11 yang digunakan 3 pin sebagai *input* yang akan mengambil nilai berupa suhu dan kelembapan, nilai tersebut digunakan sebagai data untuk menampilkan hasil data suhu dan kelembapan yang sudah diproses melalui mikrokontroler terlebih dahulu. Data dari sensor DHT11 akan ditampilkan pada LCD maupun aplikasi Blynk yang berfungsi melihat hasil data sensor.

#### 4.3.1.3 Relay

Relay digunakan untuk menjalankan fungsi logika yang dapat memberikan penundaan waktu dalam kontrol penyiraman. Relay yang digunakan terdapat 1 *channel* dengan memiliki tegangan 3.3V sampai 5V DC untuk melakukan kontrol terhadap penyiraman. Penggunaan Relay sangat baik untuk dapat melakukan *switch* pada perangkat AC maupun DC yang membutuhkan arus dan tegangan yang besar. Pada Gambar 4.7 menampilkan skematik Relay.





**Gambar 4.7 Skematik Relay**

Sumber : (AliExpress, 2018)

Prinsip kerja Relay untuk menggerakkan rangkaian koil magnet sebagai pembuka dan penutup saklar, yang membedakan antara pembuka dan penutup saklar adalah kekuatan dari saklar dalam menghubungkan besaran arus listrik yang melewatinya. Jadi, ketika kita memberikan arus listrik pada koil magnet pada Relay atau kontraktor, maka saklar akan terhubung. Saklar ini disebut sebagai kontak *Normally Open* (NO) yaitu bila koil magnet Relay atau kontraktor dalam keadaan tak terhubung arus listrik, karena dalam kondisi terbuka. Sedangkan kontak *Normally Close* (NC) yaitu koil magnet Relay atau kontraktor dalam keadaan terhubung, karena koil magnet dalam kondisi menempel. Pada Tabel 4.4 menampilkan spesifikasi Relay.

**Tabel 4.4 Spesifikasi Relay**

Sumber : (Addicore, 2018)

Fungsi	Spesifikasi
Suplai Tegangan	3.3~6V
Suplai Arus Energi	2Ma
Suplai Arus dengan relay	70-72mA
Kontrol sinyal	Active Low
Kontrol arus sinyal	1.5-1.9mA
Tegangan relay	250 VAC atau 30 VDC
Arus relay	10A

Peneliti menggunakan Relay sebagai saklar *switch on/off* sebagai pengendali dalam melakukan penyiraman ataupun tidak menyiram. Relay dianalogikan sebagai pemutus dan penghubung seperti pada fungsi tombol *push button*.

#### 4.3.1.4 Pompa Air

Pompa air digunakan sebagai elemen yang berfungsi untuk menyerap sekaligus mendorong air, air yang terdapat pada ruang *impeller* akan digerakan menggunakan sebuah motor. Selama *impeller* tersebut berputar, air akan didorong keluar menuju ke selang untuk melakukan penyiraman. Adapun pompa air yang digunakan untuk proses penyiraman dapat dilihat seperti pada Gambar 4.8.



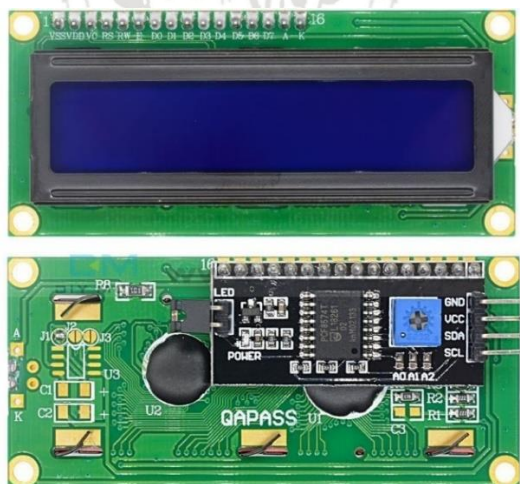
**Gambar 4.8 Pompa air mini**

Sumber : (Artofrcircuit, 2018)

Peneliti menggunakan pompa sebagai penyerap sekaligus mendorong air untuk melakukan penyiraman, jenis pompa yang menggunakan motor DC 6V dan menggunakan tegangan searah sebagai sumber tenaganya.

#### 4.3.1.5 *Liquid Crystal Display (LCD)*

LCD berfungsi untuk menampilkan nilai suhu dan kelembapan yang didapat dari sensor DHT11. LCD merupakan modul yang sangat sering digunakan pada proyek berbasis mikrokontroler, jika tidak menggunakan LCD *interface* nya tidak terlalu menarik. Pada jaman dulu kita menggunakan LCD sering memiliki masalah dengan kurangnya pin *input* maupun *output* karena rata-rata modul LCD untuk yang paling kecil saja memerlukan pin sekitar 6 sampai 8 pin, maka dengan adanya modul *interface* LCD ke I2C hanya memerlukan empat kabel untuk dapat terhubung ke LCD (Suranata, 2017). Pada Gambar 4.9 menampilkan LCD dan I2C yang sudah dirakit menjadi 1 komponen.



**Gambar 4.9 Penerapan komponen I2C pada LCD**

Sumber : (Diymore, 2018)

Pada modul *Liquid Cristal Display (LCD)* terdapat mikrokontroler yang berfungsi untuk menampilkan karakter seperti huruf, angka dan grafik pada layar. Pada bagian I2C didesain untuk dapat mengirim dan menerima data. Tabel 4.5 menjelaskan pin kaki atau jalur *input* dan kontrol pada LCD.

Tabel 4.5 Pin *input* pada I2C

Pin	Fungsi
SCL (Serial Clock)	Sinkronisasi jalur data transfer pada I2C
SDA (Serial Data)	Jalur data
VCC	5V Power Supply
GND	Ground

Pada bagian pengabelan, seperti halnya dengan modul berbasis komunikasi I2C lainnya, modul ini tentu hanya menggunakan dua kabel data, dan dua lainnya pada kabel *VCC* dan *GND*. Adapun penempatan pin *SCL* dihubungkan pada pin *D1*, pin *SDA* dihubungkan pada Pin *D2*, sedangkan pin *VCC* dihubungkan ke pin *VU* (5V), dan pin *GND* ke *GND*.

Peneliti menggunakan LCD karena cocok untuk menampilkan hasil data informasi melalui layar LCD, dengan ukuran 16x2 yang dirasa sudah cukup sesuai dengan yang dibutuhkan oleh sistem. Dengan ditambahkannya modul I2C dapat lebih mudah dalam menghubungkan pada mikrokontroler NodeMCU, karena dapat meringkas 16 pin yang disediakan LCD menjadi 4 pin seperti yang tampilan pada Tabel 4.5.

#### 4.3.1.6 LM2596

Rangkaian regulator LM2596 banyak digunakan untuk sistem kontrol seperti arduino, NodeMCU dan ATMEGA. Rangkaian modul LM2596 dapat digunakan untuk menurunkan tegangan DC maksimal 3V dengan *range* DC 3.2V-46V dengan selisih minimum I/O 1.5V. Keunggulan modul step down LM2596 besarnya tegangan *output* tetap stabil meskipun tegangan input naik turun, *output* dapat diatur dengan memutar potensiometer atau baut warna emas dengan menggunakan obeng minus. Pada Gambar 4.10 menampilkan modul *step down* LM2596.

Gambar 4.10 Modul *step down* LM2596

Sumber : (Itead, 2017)

Penggunaan modul *step down* LM2596 untuk menurunkan tegangan yang didapat dari baterai DC 9V menjadi 6V, karena dikhawatirkan ketika arus tegangan terlalu tinggi dapat membahayakan komponen yang lain.

#### 4.3.1.7 Smartphone

Smartphone berfungsi untuk menghubungkan NodeMCU ESP8266 dengan wi-fi pada *smartphone* dengan menghubungkan jaringan yang sama agar dapat terhubung dan dapat melakukan pemantauan melalui aplikasi Blynk. Menurut Sawyer (2011) definisi dari *smartphone* yaitu telepon seluler yang memakai beberapa layanan seperti mikroprosesor, memori, layar dan modem bawaan. Dengan begitu, *smartphone* memiliki fitur yang lengkap dibandingkan telepon seluler biasa. Pada Gambar 4.11 menunjukkan *smartphone* yang memiliki fitur lengkap.



**Gambar 4.11 Smartphone**

Sumber : (Beavis, 2018)

Penggunaan *smartphone* untuk memasang aplikasi Blynk yang digunakan untuk pemantauan suhu dan kelembapan pada kubung jamur dari jarak jauh. Pemantauan ini dilakukan dengan melihat hasil informasi yang didapatkan dari sensor DHT11, informasi akan ditampilkan pada layar aplikasi Blynk berupa parameter data suhu dan kelembapan.

#### 4.3.1.8 Power Bank

*Power bank* digunakan peneliti sebagai daya terhadap mikrokontroler NodeMCU, karena ketika melakukan observasi di dalam kubung jamur tidak ada stop kontak aliran listrik. Penggunaan *power bank* sangat bermanfaat karena dengan menghubungkan USB pada kabel data *handphone* android yang sama dengan port daya pada NodeMCU. Pada Gambar 4.12 menunjukkan *power bank* yang digunakan sebagai daya pada NodeMCU.



**Gambar 4.12 Power bank**

Sumber : (Terlaris.aseanpriceblog, 2018)

Peneliti menggunakan *power bank* yang memiliki kapasitas 25.000 mAh, sebagai alat bantu daya yang diharapkan ketika melakukan implementasi dapat membantu dalam melakukan pengujian sehingga tidak merepotkan petani untuk



memasang stop kontak pada kubung jamur. Pada Tabel 4.6 terdapat *input* dan *output* pada *power bank* yang digunakan peneliti.

**Tabel 4.6 Pin *input* dan *output power bank***

Fungsi	Spesifikasi
<i>Input</i>	5V/2A
<i>Output 1</i>	5V/1A
<i>Output 2</i>	5V/2A

#### 4.3.2 Kebutuhan Perangkat Lunak

Pada penelitian ini dibutuhkan beberapa perangkat lunak untuk menunjang penelitian, antara lain:

1. Microsoft Windows 10 Pro 64-bit  
Perangkat lunak ini digunakan sebagai sistem operasi untuk melakukan aktivitas dalam pembuatan sistem melalui laptop.
2. Aplikasi Blynk  
Perangkat lunak ini digunakan untuk melakukan pemantauan dengan menggunakan *smartphone*.
3. Arduino *Integrated Development Environment* (IDE)  
Perangkat lunak ini digunakan untuk menuliskan *source code* program pada *sketch* dan juga terdapat *board manager* untuk inialisasi *library* yang diterapkan.

#### 4.3.3 Kebutuhan Fungsional

Sistem mempunyai kebutuhan fungsional yang berisi proses-proses harus terpenuhi yang akan dilakukan oleh sistem, antara lain :

1. Modul NodeMCU dapat membaca data dari sensor DHT11
2. NodeMCU dapat mengontrol dan mengakuisisi data I/O.
3. Pemantauan dapat dilakukan dari jauh
4. Blynk dan LCD menampilkan hasil data pemantauan sesuai dengan kondisi suhu dan kelembapan pada ruangan kubung jamur
5. Penyiraman dapat bekerja sesuai dengan *rule-rule fuzzy* yang sudah ditentukan

#### 4.3.4 Kebutuhan Non-fungsional

Kebutuhan non-fungsional menjelaskan apa saja yang menjadi batasan dalam kebutuhan desain sistem untuk memudahkan pengguna dalam mengoperasikan sistem, antara lain:

1. Kubung jamur

Penggunaan kubung jamur dilakukan peneliti sebagai kebutuhan dalam proses untuk merancang dan mengimplementasikan sistem pemantauan yang akan diterapkan peneliti.

2. Air

Air sangat dibutuhkan untuk melakukan implementasi oleh peneliti, karena digunakan sebagai sarana penyiraman tanaman jamur.

#### 4.3.5 Batasan Desain Sistem

Dalam pembuatan sistem pemantauan menggunakan Blynk dan pengendalian penyiraman tanaman jamur dengan metode logika *fuzzy* terdapt beberapa batasan agar sistem ini dapat diterapkan sesuai dengan harapan peneliti, maka perlu diterapkan beberapa batasan-batasan kebutuhan sistem, anantara lain:

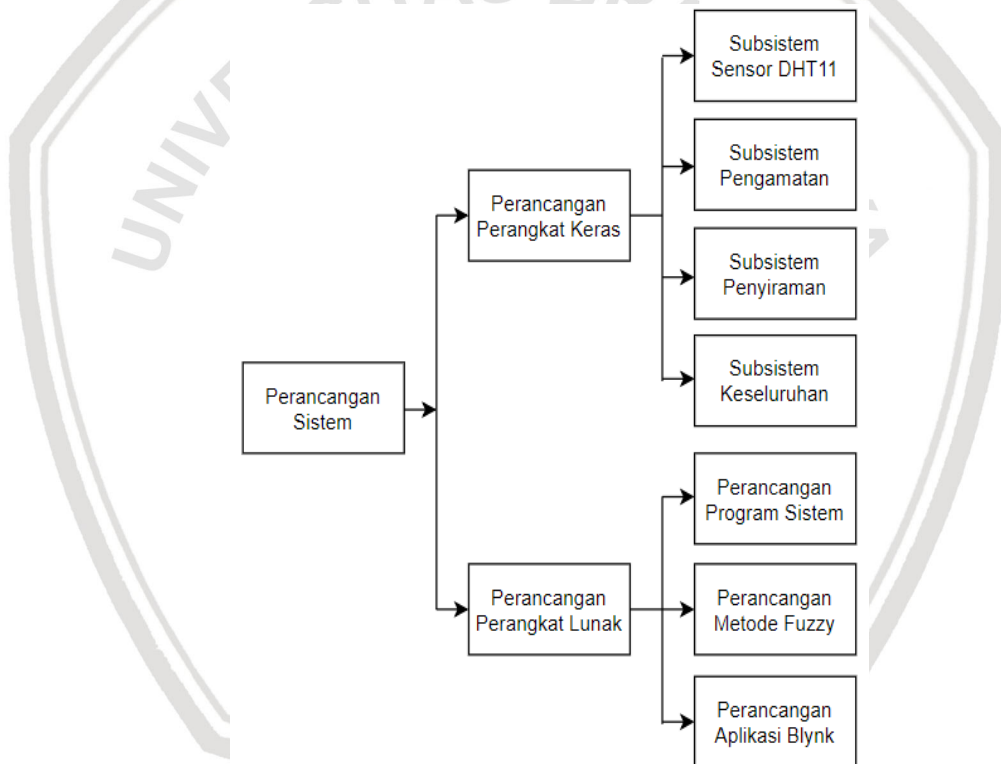
1. Sensor DHT11 hanya memiliki jangkauan beberapa meter.
2. Sensor DHT11 digunakan untuk mendeteksi suhu dan kelembapan.
3. LCD hanya menampilkan nilai suhu dan kelembapan.
4. Aplikasi Blynk hanya menampilkan nilai suhu dan kelembapan.
5. Metode logika *fuzzy* dipilih sebagai eksekusi *rule-rule fuzzy* yang sudah ditentukan untuk melakukan penyiraman.

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini penulis akan menjelaskan bentuk perancangan dan implementasi sistem yang akan dikerjakan. Perancangan sistem dan implementasi sistem dibagi menjadi 2 bagian yaitu perancangan perangkat keras dan perancangan perangkat lunak.

### 5.1 Perancangan Sistem

Perancangan sistem dilakukan setelah melakukan analisis kebutuhan sistem. Perancangan ini dilakukan untuk merencanakan tahapan-tahapan yang harus dikerjakan penulis dalam memulai proses perancangan. Tahapan penelitian yang harus dilakukan antara lain perancangan perangkat keras dan perangkat lunak. Diharapkan dengan dilakukannya tahapan perancangan sistem dapat memudahkan proses pelaksanaan pengerjaan. Pada Gambar 5.1 menampilkan diagram perancangan sistem.



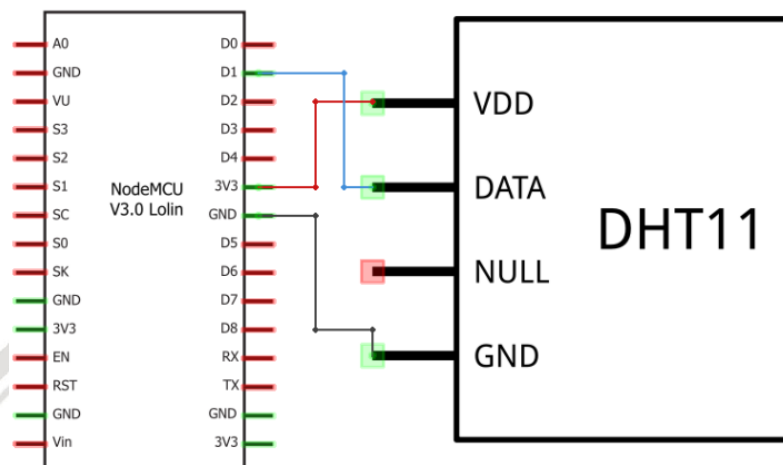
Gambar 5.1 Blok diagram perancangan sistem

#### 5.1.1 Perancangan Perangkat Keras

Perancangan perangkat keras pada sistem ini pertama-tama dengan menghubungkan komponen seperti sensor, LCD, Relay, pompa air, LM2596 dan baterai. Lebih jelasnya perancangan perangkat keras dibagi menjadi 4 subsistem yaitu perancangan subsistem sensor DHT11, perancangan subsistem pemantauan, perancangan subsistem penyiraman dan perancangan subsistem keseluruhan.

#### 5.1.1.1 Perancangan Subsistem Sensor DHT11

Perancangan subsistem sensor DHT11 dilakukan dengan menghubungkan data, tegangan dan *ground* pada NodeMCU, dari perancangan subsistem ini diharapkan dapat berjalan sesuai dengan rencana agar dapat diimplementasikan pada kubung jamur untuk mendapatkan nilai suhu dan kelembapan. Pada Gambar 5.2 menampilkan perancangan subsistem sensor DHT11.



**Gambar 5.2 Subsistem sensor DHT11**

Perancangan skematik pada Gambar 5.2 menunjukkan susunan pin-pin yang terhubung antara sensor DHT11 dengan NodeMCU. Pada perancangan ini peneliti menggunakan sensor DHT11 dengan 3 pin, karena ada juga sensor DHT11 dengan menggunakan 4 pin, akan tetapi salah satu pinnya tidak digunakan. Pada Tabel 5.1 menampilkan konfigurasi pin-pin yang digunakan.

**Tabel 5.1 Konfiguasi pin sensor DHT11 dengan NodeMCU**

Sensor DHT11	NodeMCU
OUT	D1
VCC	3V
GND	GND

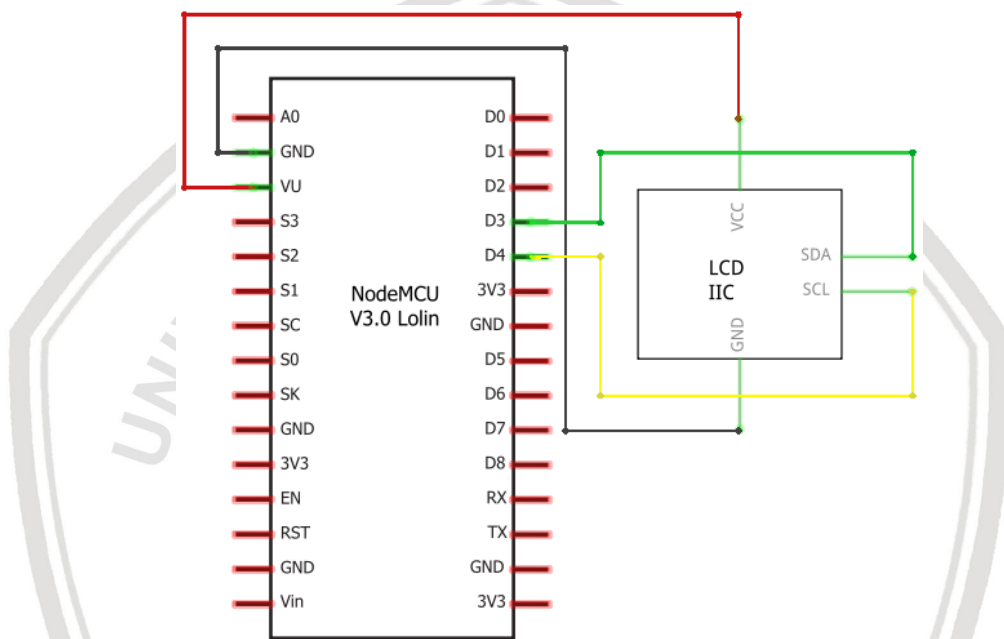
Pada Tabel 5.1 merupakan konfigurasi perancangan pin sensor DHT11 dengan mikrokontroler NodeMCU, pin *out* pada sensor DHT11 adalah pin data yang akan berguna untuk mengirimkan sinyal hasil data sensor ke pin *D1* NodeMCU. Pin *D1* merupakan pin digital untuk mengeset pin agar dapat menerima masukan sinyal sebagai penerima hasil data *output* dari sensor. Sinyal data bit yang diterima akan di proses oleh mikrokontroler NodeMCU dengan memproses 16 bit mewakili data pertama suhu relatif, 16 bit mewakili kelembapan relatif dan 8 bit terakhir merupakan *checksum*. Kemudian nilai biner 16 bit dari suhu dan kelembapan di rubah menjadi nilai desimal untuk ditampilkan pada *serial monitor*, sedangkan nilai *checksum* untuk mengkonfirmasi data yang diterima tidak mengandung kesalahan dengan menyesuaikan SUM dari empat bit pertama (8 bit) dengan bit terakhir pada *checksum*. Setelah itu pin VCC dihubungkan dengan pin 3V pada



NodeMCU karena sensor DHT11 hanya membutuhkan tegangan 3V dan pin *ground* pada sensor dihubungkan dengan pin *ground* pada NodeMCU. Jumlah sensor yang digunakan pada sistem ini menggunakan 1 sensor dengan output 2 parameter yang dipasang pada bagian tengah sistem.

#### 5.1.1.2 Perancangan Subsistem Pengamatan

Perancangan subsistem pengamatan digunakan untuk melihat hasil nilai yang didapat dari sensor DH11. Hasil nilai ditampilkan pada layar LCD maupun aplikasi Blynk, data hasil sensor digunakan sebagai antarmuka dalam melakukan pemantauan. Pada Gambar 5.3 menunjukkan perancangan subsistem pemantauan dengan menghubungkan pin yang ada pada I2C ke pin NodeMCU.



**Gambar 5.3 Perancangan subsistem LCD**

Pada perancangan skematik pada Gambar 5.3 merupakan perancangan subsistem LCD yang sudah diintegrasikan dengan modul I2C. Peneliti menggunakan modul I2C karena dapat meringkas pin-pin pada LCD menjadi 4 pin yaitu pin VCC, GND, SDA dan SCL. Pada Tabel 5.2 menampilkan konfigurasi antara pin I2C ke pin NodeMCU.

**Tabel 5.2 Konfigurasi I2C dengan NodeMCU**

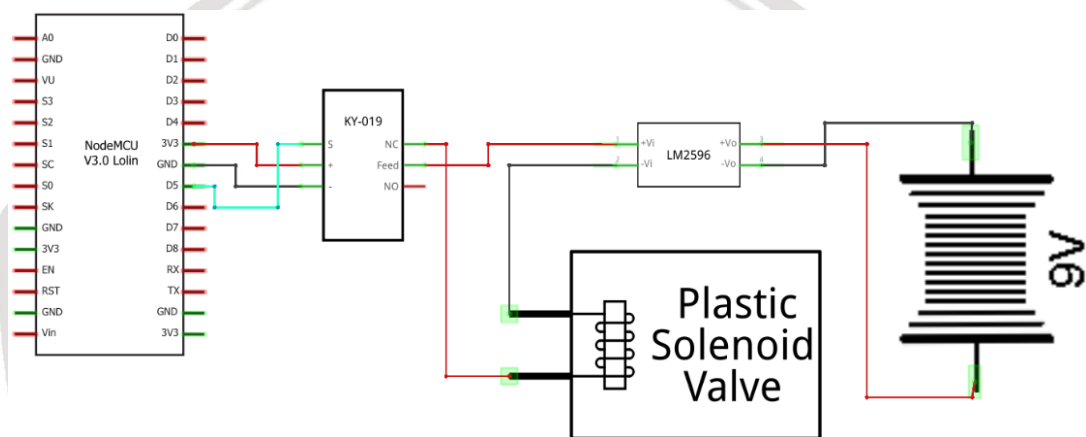
LCD (I2C)	NodeMCU
GND	GND
VCC	VU(5V)
SDA	D3
SCL	D4

Pada Tabel 5.2 konfigurasi pin *ground* pada I2C dihubungkan dengan pin *ground* NodeMCU, pin VCC pada I2C dihubungkan dengan pin VU pada NodeMCU,

pin *VU* merupakan port yang memiliki tegangan 5V pada NodeMCU yang dibutuhkan sebagai tegangan pada LCD, pin *SDA* dihubungkan dengan pin *D3* pada NodeMCU yang digunakan sebagai pengiriman *serial* data sedangkan pin *SCL* dihubungkan dengan pin *D4* yang berfungsi sebagai *serial clock* untuk menyesuaikan pengiriman data antara perangkat pada bus I2C yang dihasilkan oleh perangkat *master*. Pada modul I2C setiap perangkat memiliki ID *preset* atau alamat perangkat yang unik sehingga master dapat memilih perangkat mana yang akan digunakan untuk berkomunikasi.

### 5.1.1.3 Perancangan Subsistem Penyiraman

Perancangan pada subsistem penyiraman dilakukan dengan merancang beberapa komponen seperti baterai 9V, step down LM2596, modul relay, dan pompa air sebagai alat bantu penyiraman. Pada Gambar 5.4 menunjukkan hasil perancangan subsistem penyiraman.



**Gambar 5.4 Perancangan subsistem penyiraman**

Pada Gambar 5.4 merupakan perancangan pada subsistem penyiraman, peneliti menggunakan relay sebagai kontrol pompa dalam melakukan penyiraman, LM2596 digunakan untuk menurunkan tegangan yang dikirimkan baterai agar tidak terlalu tinggi. Pada Tabel 5.3 menampilkan konfigurasi pin-pin yang terhubung dalam perancangan subsistem penyiraman.

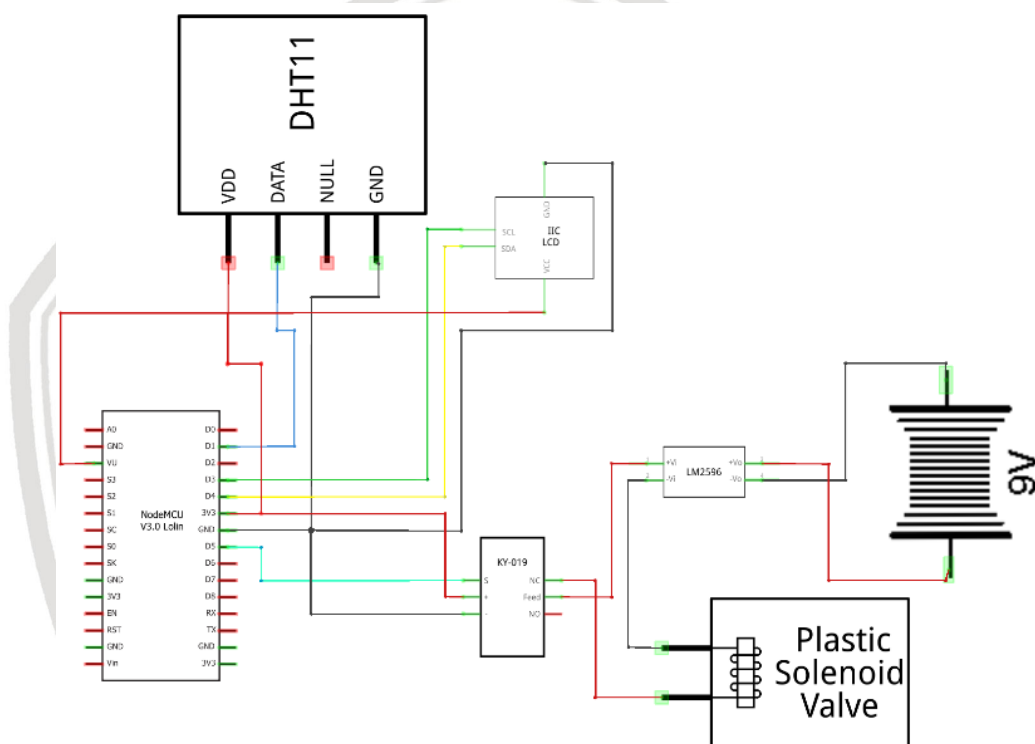
**Tabel 5.3 Konfigurasi pin subsistem penyiraman**

Baterai 9V	LM2596		Pompa Air	Relay		NodeMCU
Pin +	IN +	OUT +	-	COM	VCC	3V
-	-	-	OUT +	NC	GND	GND
Pin -	IN -	OUT -	OUT -	NO (null)	IN	D5

Pada Tabel 5.3 menghubungkan antara konfigurasi pin baterai, *step down* LM2596 DC, Pompa air, single relay, dan NodeMCU. Baterai memiliki 2 porta yaitu porta positif dan porta negatif yang dihubungkan ke *step down* LM2596, fungsi dari LM2596 untuk menurunkan tegangan dari baterai agar tidak terlalu tinggi,

sehingga tegangan diubah menjadi 6V. Porta tegangan positif pada baterai dihubungkan ke porta *in positif* LM2596 dan porta negatif nya dihubungkan ke porta *in negatif* LM2596. Kemudian porta *out* positif LM2596 dihubungkan ke relay pada porta COM, porta *out* negatif dihubungkan ke porta negatif pompa air, pompa menghubungkan porta positif ke pin *Normally Close*. Setelah itu relay menghubungkan pin *VCC* ke NodeMCU melalui pin 3V karena dengan tegangan 3V relay sudah bisa bekerja dengan baik, *ground* dihubungkan ke ground dan pin *IN* dihubungkan ke pin D5 pada NodeMCU sebagai masukan data. Saklar pada Relay terjadi perubahan *switch on/off* ketika memperoleh energi elektro magnetik pada *armature* Relay, maka *switch* akan bergerak untuk menerapkan tegangan ke Relay.

#### 5.1.1.4 Perancangan Komponen Keseluruhan



**Gambar 5.5 Komponen keseluruhan**

Pada Gambar 5.5 digabungkan seluruh komponen subsistem perancangan yang sudah dibuat. Dengan menggabungkan seluruh subsistem perancangan diharapkan sistem sudah bisa diimplementasikan dan bisa melakukan eksekusi ketika sudah dinyalakan. Hubungan pin *input* dan *output* sudah diterapkan sesuai dengan yang dijelaskan pada subsistem sensor DHT11, subsistem pengamatan dan subsistem penyiraman. Setelah perancangan sudah siap, maka dilakukan pengujian yang akan dijelaskan pada bab pengujian dan analisis.

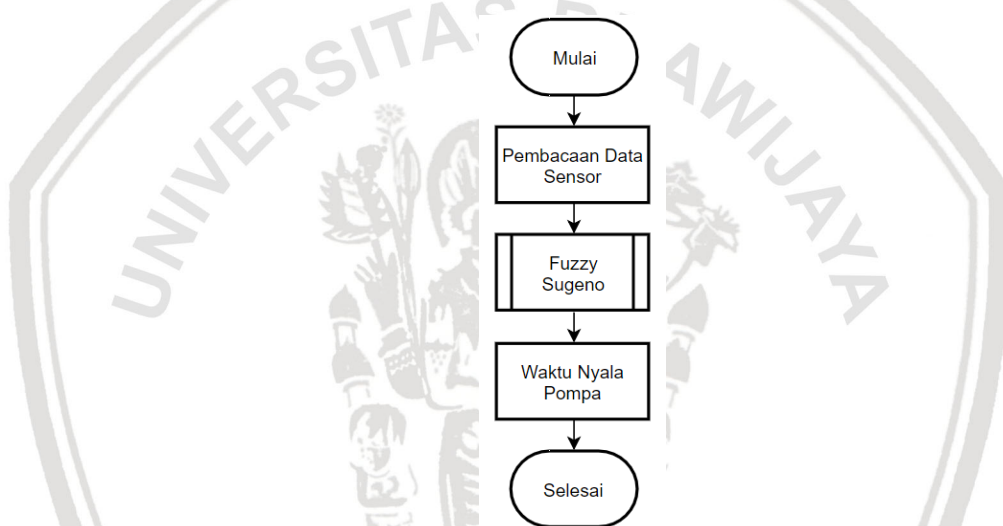
#### 5.1.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada sistem ini dibagi menjadi 3 pembahasan yaitu perancangan program sistem, perancangan metode *fuzzy* dan perancangan

aplikasi Blynk. Dari ketiga pembahasan tersebut akan dijelaskan secara rinci pembuatan perancangan perangkat lunak yang dikerjakan.

#### 5.1.2.1 Perancangan Program Sistem

Pada perancangan program sistem dibuat rangkaian yang disusun menjadi satu sebagai langkah dalam menyelesaikan masalah. Kemudian hasil tersebut diimplementasikan menggunakan Bahasa pemrograman sehingga ketika sudah melakukan *compile*, program bisa dieksekusi oleh mikrokontroler. Mula-mula program menginisialisasi *library* dari sensor, aktuator dan *fuzzy*, inisialisasi *library* diletakkan sebelum program *void setup*. Pada bagian *void setup* berisi perintah untuk memanggil inisialisasi dari masing-masing *library*. Kemudian pada bagian *void loop* berisi perintah *output* program dan juga perintah logika *array* yaitu *if else* sebagai perintah *looping* agar program terus berjalan untuk menentukan eksekusi perintah yang tepat. Pada Gambar 5.6 menunjukkan perancangan diagram alir program sistem.



**Gambar 5.6 Perancangan diagram alir program sistem**

Pada Gambar 5.6 merupakan langkah-langkah yang ada dilakukan pada program sistem. Langkah-langkah di atas berfungsi untuk menjelaskan hasil pembacaan data sensor yang akan diproses dengan menggunakan metode *fuzzy Sugeno*, sampai melakukan keluaran berupa nyala pompa. Uraian perancangan diagram program sistem dijelaskan sebagai berikut:

1. Pembacaan Data Sensor

Masukan yang didapat dari sensor DHT11 dikirimkan ke mikrokontroler untuk diproses. Pada bagian proses tersebut mikrokontroler melakukan pembacaan data sensor.

2. *Fuzzy Sugeno*

Metode logika *fuzzy* yang dipilih yaitu model Takagi Sugeno. Pemanggilan fungsi *fuzzy Sugeno* untuk menentukan apakah kondisi kubung jamur butuh penyiraman atau tidak.



### 3. Waktu Nyala Pompa

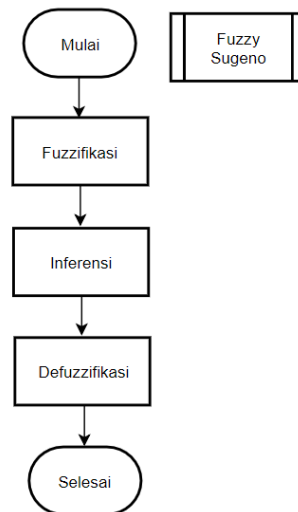
Pompa akan melakukan penyiraman apabila *rule fuzzy* mendeteksi kondisi suhu dan kelembapan pada kubung jamur membutuhkan penyiraman. Durasi nyala pompa *off delay* selama 2 detik, penyiraman sedikit selama 7,5 detik dan penyiraman banyak selama 15 detik.

#### 5.1.2.2 Perancangan Metode Fuzzy

Pada saat melakukan diskusi dengan Ibu Zunaini penyuluh tanaman jamur pada Balai Pengkajian Teknologi Pertanian, Karangploso, Malang. Tanaman jamur membutuhkan penyiraman yang dilakukan pada lantai ruangan agar suhu dan kelembapan ruangan tetap dalam kondisi stabil yaitu suhu 22-26°C dan kelembapan 80-90%. Penempatan kubung jamur dibangun di bawah pohon sehingga mengurangi pancaran sinar matahari secara langsung. Penyiraman dilakukan apabila kelembapan kurang dari 80% dan tidak dilakukan penyiraman apabila lebih dari 90%, akan tetapi di dalam kubung jamur hanya ada termometer untuk mendeteksi suhu ruangan. Suhu yang digunakan untuk penyiraman yaitu ketika suhu lebih dari 26°C. penyiraman pada musim hujan dilakukan 1 kali/hari, sedangkan pada musim panas penyiraman dilakukan 3 kali/hari. Jadi, pada kubung jamur hanya melihat kondisi suhu untuk melakukan penyiraman, sirkulasi udara pada kubung jamur diperoleh dari lubang penyekat yang terbuat dari anyaman bambu dan juga dari jendela.

Metode *fuzzy* dirancang dengan menggunakan bentuk trapesium karena pada fungsi trapesium terdapat beberapa nilai  $x$  yang akan diubah dalam bilangan *fuzzy*, fungsi ini memiliki 2 dari 4 nilai domain terkecil dan terbesar yang bernilai 1, Sehingga nilai domain terkecil dan terbesar lainnya bernilai 0. Angka yang diletakkan pada setiap set trapesium merupakan vektor bilangan yang didapat dari jumlah volume suhu. Suhu dapat dikatakan dingin ketika 18-22°C, sedang 22-26°C dan panas 26-30°C, sedangkan jumlah volume kelembapan yaitu kering 75-80%, normal 80-90% dan basah 90-95% nilai didapat dari wawancara secara langsung dengan peneliti tanaman jamur. Alasan menentukan fungsi keanggotaan himpunan *fuzzy* dengan menggunakan 3 elemen yang ditentukan pada nilai keanggotaan trapesium sebagai keseluruhan nilai yang harus dihitung. Dengan semakin sedikit himpunan keanggotaan yang digunakan dapat mengurangi *error* pada saat eksekusi dilakukan.

Dapat dilihat pada Gambar 5.7 menampilkan perancangan metode *fuzzy* untuk menentukan *rule* penyiraman. Logika *fuzzy* memberikan efisiensi lebih ketika diterapkan dalam sistem kontrol, karena mempunyai konsep matematis yang sederhana dan mudah dimengerti. Penerapan metode *fuzzy* menggunakan model *Takagi Sugeno*, model ini dikembangkan sebagai pendekatan secara sistematis untuk menghasilkan *rule fuzzy* dari hasil data sensor. Sebelum melakukan eksekusi penyiraman, terdapat beberapa tahap yaitu fuzzifikasi, inferensi dan defuzzifikasi.



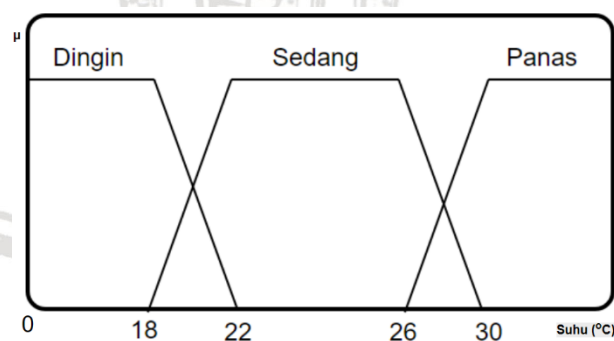
**Gambar 5.7 Perancangan diagram alir metode fuzzy**

Uraian dari tahapan perancangan diagram alir metode *fuzzy* yaitu sebagai berikut:

### 1. Fuzzifikasi

Dalam perancangan logika *fuzzy*, langkah pertama yang dilakukan adalah proses fuzzifikasi yaitu proses perubahan nilai tegas (*crisp*) dalam arti keanggotaan *fuzzy* untuk menentukan derajat keanggotaan variabel *input fuzzy* yaitu derajat keanggotaan sensor suhu dan kelembapan. Pada proses fuzzifikasi jika suatu suhu atau kelembapan mengalami perubahan maka proses fuzzifikasi dapat mendeteksi *rule* yang akan di terapkan pada aktuator.

Fuzzifikasi dimulai dari menerima *input* sensor, yang terdiri dari sensor suhu dan kelembapan dan data tersebut digunakan untuk menghitung derajat keanggotaan nilai dari sensor, kemudian hasilnya akan diperoleh derajat himpunan berupa nilai suhu dan kelembapan.



**Gambar 5.8 Fungsi himpunan suhu**

Gambar 5.8 merupakan fungsi himpunan sensor suhu. Terdapat 3 himpunan keanggotaan, dingin suhu kurang dari 22°C, sedang yaitu suhu antara 22°C-26°C dan panas suhu lebih dari 26°C. Dari gambar tersebut dapat dimasukkan ke dalam representasi kurva trapesium untuk mencari derajat keanggotaan suhu sehingga didapatkan perhitungan sebagai berikut:

- Dingin [0 22]

$$\mu[\text{Dingin}] = \begin{cases} 1, & x \leq 18 \\ \frac{22-x}{22-18}, & x > 18 \text{ atau } x \leq 22 \\ 0, & x > 22 \end{cases}$$

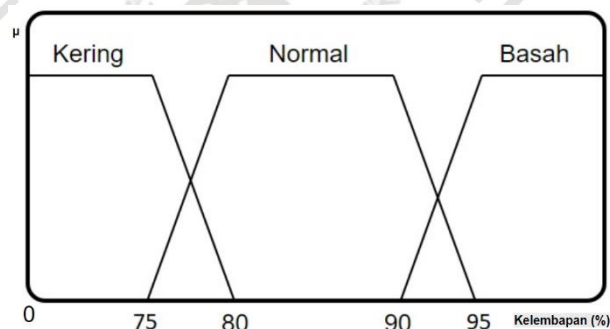
- Sedang [22 26]

$$\mu[\text{Sedang}] = \begin{cases} 0, & x \leq 18 \\ \frac{x-18}{22-18}, & x > 18 \text{ atau } x \leq 22 \\ 1, & 22 < x \leq 26 \\ 0, & x \geq 30 \end{cases}$$

- Panas [26 30]

$$\mu[\text{Panas}] = \begin{cases} 0, & x \leq 26 \\ \frac{x-26}{30-26}, & x > 26 \text{ atau } x \leq 30 \\ 1, & x \geq 30 \end{cases}$$

Adapun fungsi dari himpunan bilangan himpunan kelembapan yaitu sebagai berikut:



**Gambar 5.9 Fungsi himpunan kelembapan**

Gambar 5.9 merupakan fungsi himpunan kelembapan. Terdapat 3 himpunan keanggotaan, kering kelembapan kurang dari 80%, normal kelembapan antara 80-90% dan basah kelembapan lebih dari 90%. Dari gambar tersebut dapat dimasukkan ke dalam *representase* kurva trapesium untuk mencari derajat keanggotaan kelembapan sehingga didapatkan perhitungan sebagai berikut:

- Kering [0 80]

$$\mu[\text{Kering}] = \begin{cases} 1, & y \leq 75 \\ \frac{80-y}{80-75}, & y > 75 \text{ atau } y \leq 80 \\ 0, & y > 80 \end{cases}$$

- Normal [80 90]

$$\mu[\text{Normal}] = \begin{cases} 0, & y \leq 75 \\ \frac{y-75}{80-75}, & y \leq 75 \text{ atau } y \geq 90 \\ 1, & 80 > y \leq 90 \\ 0, & y \geq 95 \end{cases}$$

- Basah[90 95]

$$\mu[\text{Basah}] = \begin{cases} 0, & y \leq 90 \\ \frac{y-90}{95-90}, & y > 90 \text{ atau } y \leq 95 \\ 1, & y \geq 95 \end{cases}$$

## 2. Inferensi

Setelah proses fuzzifikasi dilakukan, maka tahap selanjutnya yaitu proses inferensi untuk mengevaluasi *rule fuzzy* agar didapatkan hasil *output* dari tiap *rule*. Pada Tabel 5.4 menampilkan hasil *output* dari tiap *rule fuzzy*.

**Tabel 5.4 Rule fuzzy**

Suhu	Kelembapan	Rule
Dingin	Kering	Rule 0
Dingin	Normal	Rule 1
Dingin	Basah	Rule 2
Sedang	Kering	Rule 3
Sedang	Normal	Rule 4
Sedang	Basah	Rule 5
Panas	Kering	Rule 6
Panas	Normal	Rule 7
Panas	Basah	Rule 8

Pada Tabel 5.4 terdapat 9 *rule fuzzy* yang didapat dari data keanggotaan himpunan pada tahap fuzzifikasi. Kemudian hasil dari *rule fuzzy* diteruskan ke tahap selanjutnya yaitu defuzzifikasi sebagai tahap akhir perhitungan *crisp output*.

## 3. Defuzzifikasi

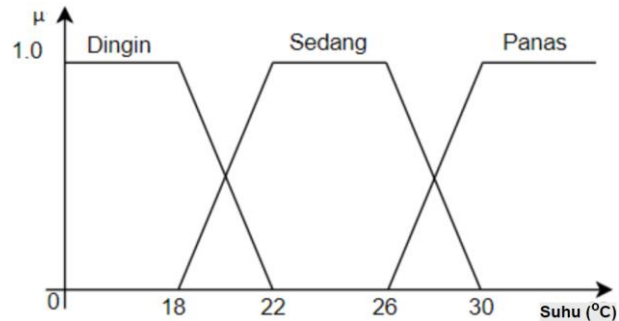
Setelah proses inferensi selesai, maka proses selanjutnya adalah defuzzifikasi. Proses defuzzifikasi merupakan salah satu proses pemetaan himpunan *fuzzy* ke dalam himpunan tegas (*crisp*) dengan cara mencari nilai maksimum. Setelah mendapatkan nilai  $\alpha$  dari metode *Min-Max* pada setiap variabel akan dievaluasi pada setiap *rule* atau aturan untuk menentukan nilai *crisp output*. Defuzzifikasi pada sistem ini memiliki 3 bagian penyiraman yaitu *off*, sedikit dan banyak.

Keanggotaan *off* merupakan perintah untuk tidak melakukan penyiraman dengan *delay* 2 detik, selanjutnya penyiraman sedikit merupakan perintah untuk melakukan penyiraman selama 7,5 detik dan penyiraman banyak merupakan perintah penyiraman selama 15 detik.

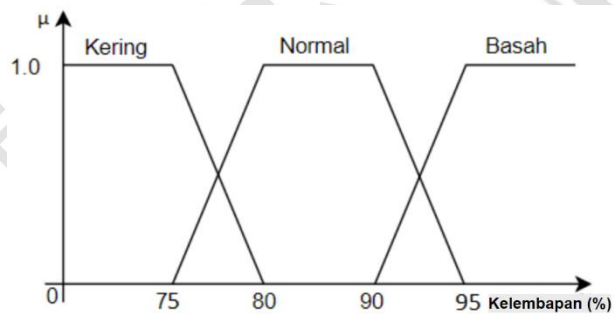
Adapun perhitungan manual *fuzzy*, diketahui *input* suhu 20°C dan kelembapan 76%. Setelah itu dilakukan tahap fuzzifikasi untuk menentukan himpunan derajat keanggotaan *crisp input* pada himpunan *fuzzy*. Pada himpunan suhu terdapat 3



kondisi linguistik yaitu dingin, sedang dan panas serta bilangan numerik suhu yaitu 18,22,26 dan 30 seperti pada Gambar 5.10. Sedangkan pada himpunan kelembapan terdapat 3 kondisi linguistik yaitu kering, normal dan basah serta bilangan numerik kelembapan yaitu 75, 80, 90 dan 95 seperti pada Gambar 5.11.

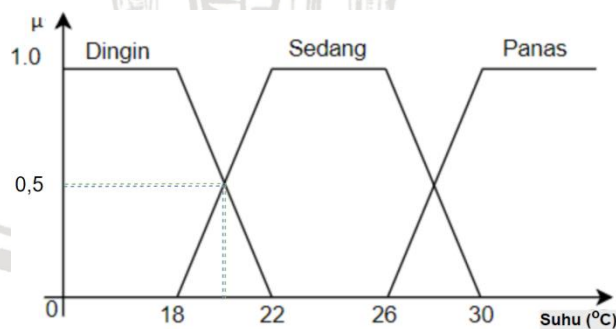


Gambar 5.10 Fungsi himpunan suhu



Gambar 5.11 Fungsi himpunan kelembapan

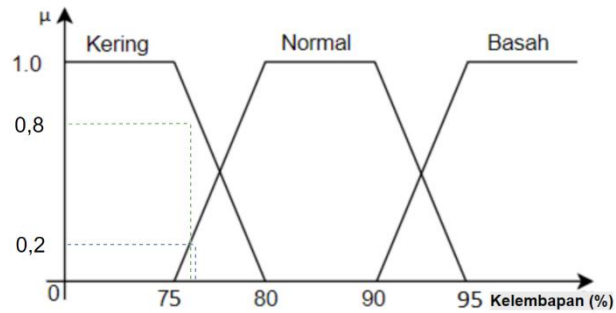
Setelah tahap fuzzifikasi selesai dikerjakan kemudian dilakukan tahap selanjutnya yaitu inferensi sebagai *evaluation rule fuzzy* untuk menghasilkan output dari tiap *rule* dan juga kombinasi dari keluaran semua *rule* seperti pada Gambar 5.12 dan Gambar 5.13.



Gambar 5.12 Rules evaluation suhu

$$\mu_{Dingin}[20] = (22-20)/(22-18) = 0,5$$

$$\mu_{Sedang}[20] = (20-18)/(22-18) = 0,5$$



**Gambar 5.13 Rule evaluation kelembapan**

$$\mu_{\text{Kering}}[76] = (80 - 76) / (80 - 75) = 0,8$$

$$\mu_{\text{Normal}}[76] = (76 - 75) / (80 - 75) = 0,2$$

Dari definisi *rule fuzzy*, didapatkan  $3 \times 3 = 9$  *rule fuzzy*, yaitu:

*Rules* [0]: IF (SUHU DINGIN && KELEMBAPAN KERING)

$$\begin{aligned} a &= \min(\mu_{\text{SUHU DINGIN}} \cap \mu_{\text{KELEMBABAN KERING}}) \\ &= \min(0,5 \cap 0,8) \\ &= 0,5 \end{aligned}$$

*Rules* [1]: IF (SUHU DINGIN && KELEMBAPAN NORMAL)

$$\begin{aligned} a &= \min(\mu_{\text{SUHU DINGIN}} \cap \mu_{\text{KELEMBABAN NORMAL}}) \\ &= \min(0,5 \cap 0,2) \\ &= 0,2 \end{aligned}$$

*Rules* [2]: IF (SUHU DINGIN && KELEMBAPAN BASAH)

$$\begin{aligned} a &= \min(\mu_{\text{SUHU DINGIN}} \cap \mu_{\text{KELEMBABAN BASAH}}) \\ &= \min(0,5 \cap 0) \\ &= 0 \end{aligned}$$

*Rules* [3]: IF (SUHU SEDANG && KELEMBAPAN KERING)

$$\begin{aligned} a &= \min(\mu_{\text{SUHU SEDANG}} \cap \mu_{\text{KELEMBABAN KERING}}) \\ &= \min(0,5 \cap 0,8) \\ &= 0,5 \end{aligned}$$

*Rules* [4]: IF (SUHU SEDANG && KELEMBAPAN NORMAL)

$$\begin{aligned} a &= \min(\mu_{\text{SUHU SEDANG}} \cap \mu_{\text{KELEMBABAN NORMAL}}) \\ &= \min(0,5 \cap 0,2) \\ &= 0,2 \end{aligned}$$

*Rules* [5]: IF (SUHU SEDANG && KELEMBAPAN BASAH)

$$a = \min(\mu_{\text{SUHU SEDANG}} \cap \mu_{\text{KELEMBABAN BASAH}})$$

$$= \min(0,5 \cap 0)$$

$$= 0$$

Rules [6]: IF (SUHU PANAS && KELEMBAPAN KERING)

$$a = \min(\mu_{SUHU PANAS} \cap \mu_{KELEMBAPAN KERING})$$

$$= \min(0 \cap 0,8)$$

$$= 0$$

Rules [7]: IF (SUHU PANAS && KELEMBAPAN NORMAL)

$$a = \min(\mu_{SUHU PANAS} \cap \mu_{KELEMBAPAN NORMAL})$$

$$= \min(0 \cap 0,2)$$

$$= 0$$

Rules [8]: IF (SUHU PANAS && KELEMBAPAN BASAH)

$$a = \min(\mu_{SUHU PANAS} \cap \mu_{KELEMBAPAN BASAH})$$

$$= \min(0 \cap 0)$$

$$= 0$$

**Tabel 5.5 Definisi rule fuzzy**

Suhu				
Kelembapan		Dingin	Sedang	Panas
	Kering	Sedikit	Sedikit	Banyak
	Normal	Off	Off	Sedikit
	Basah	Off	Off	Off

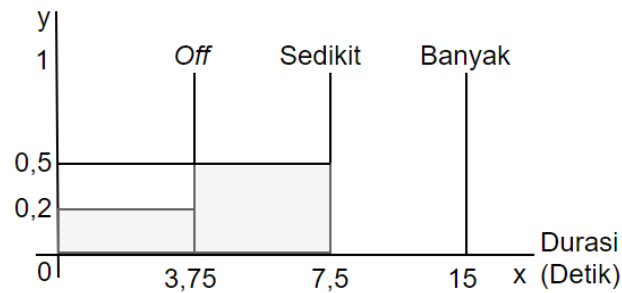
Tabel 5.5 definisi rule fuzzy terdapat empat data fuzzy input, dingin (0,5), sedang (0,5), kering (0,8) dan normal (0,2), maka didapatkan empat aturan yang dapat diaplikasikan seperti di bawah ini.

- IF (SUHU DINGIN(0,5) & KELEMBAPAN KERING(0,8))  
THEN Penyiraman is SEDIKIT 0,5
- IF (SUHU DINGIN(0,5) & KELEMBAPAN NORMAL(0,2))  
THEN Penyiraman is OFF 0,2
- IF (SUHU SEDANG(0,5) & KELEMBAPAN KERING(0,8))  
THEN Penyiraman is SEDIKIT 0,5
- IF (SUHU SEDANG(0,5) & KELEMBAPAN NORMAL(0,2))  
THEN Penyiraman is OFF 0,2

Dengan demikian diperoleh dua pernyataan yaitu:

- PENYIRAMAN OFF (0,2) OR (0,2) dihasilkan penyiraman off 0,2

- PENYIRAMAN SEDIKIT (0,5) OR (0,5) dihasilkan penyiraman sedikit 0,5



**Gambar 5.14 Fungsi *singleton* durasi penyiraman**

Kemudian tahap selanjutnya defuzzifikasi untuk menentukan hasil akhir *crisp output fuzzy*, diperoleh  $f=\{off, sedikit\}=\{0,2, 0,5\}$ . Jika proses defuzzifikasi menggunakan *high method*, maka dari 2 *fuzzy set*, *off* dan sedikit dipilih nilai minimumnya yaitu 0,5. Karena nilai *crisp* untuk penyiraman sedikit adalah 7,5 maka proses defuzzifikasi menghasilkan nilai *crisp* sebesar 7,5. Dengan demikian durasi penyiraman 7,5 detik jika dengan menggunakan *weight average* untuk defuzzifikasi diperoleh hasil.

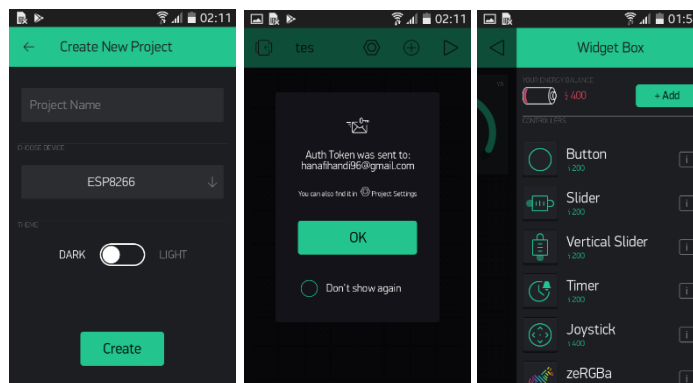
$$\begin{aligned} \text{Weight average} &= \frac{0,2 \times (3,75) + 0,5 \times (7,5)}{0,2 + 0,5} \\ &= \frac{0,75 + 3,75}{0,7} \\ &= \frac{4,5}{0,7} \\ &= 6,428 \end{aligned}$$

Dengan demikian, jika menggunakan model *Sugeno* dengan defuzzifikasi berupa *weight average*, hasil perhitungan yaitu 6,428. Hasil yang diperoleh dikategorikan pada penyiraman sedikit.

### 5.1.2.3 Perancangan Aplikasi Blynk

Pada perancangan aplikasi Blynk pertama-tama Blynk meminta *user* untuk melakukan *registrasi*, selanjutnya *user* membuat project baru dengan memilih *device* ESP8266 sebagai penghubung ke perangkat mikrokontroler NodeMCU yang sudah terintegrasi modul *wi-fi* ESP8266. Setelah itu Blynk mengirim kode autentikasi token ke *email user*. Pada Gambar 5.15 menampilkan notifikasi pengiriman *auth token* ke *email user* dan Gambar 5.16 menampilkan kode token yang dikirim ke *email*.





(a) Create Project (b) Auth Token (c) Widget Box

**Gambar 5.15 Perancangan widget Blynk**



**Gambar 5.16 Email menerima autentikasi token**

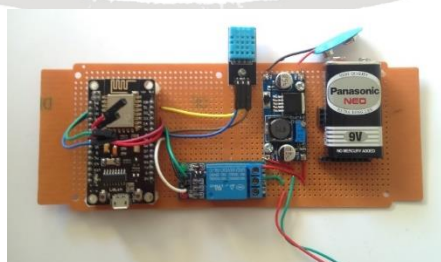
Pembuatan proyek baru dengan menggunakan *device* ESP8266 karena NodeMCU sudah terintegrasi modul *wi-fi* ESP8266. Autentikasi token yang dikirimkan ke *email* berfungsi untuk konfigurasi pada penulisan program, setelah sudah ditulis kemudian membuat penulisan program variabel dengan menuliskan nama *wi-fi* dan *password* sebagai konfigurasi *id* dan *password* pada akses poin. *Drag* dan *drop* widget yang akan digunakan sebagai data hasil informasi pemantauan yaitu *temperature* dan *humidity*. Perancangan Blynk dibuat untuk memudahkan petani dalam memantau kondisi suhu dan kelembapan kubung jamur dari jauh.

## 5.2 Implementasi Sistem

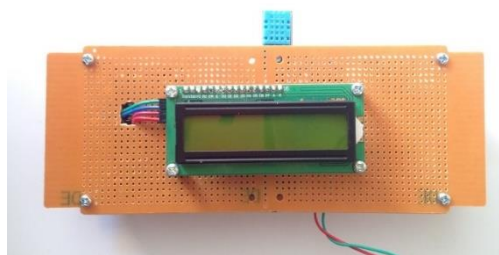
Setelah melalui perancangan sistem, maka selanjutnya adalah melakukan implementasi sistem dengan menggunakan perangkat yang sudah dirancang. Implementasi sistem yang dilakukan adalah implementasi perangkat keras dan implementasi perangkat lunak.

### 5.2.1 Implementasi Perangkat Keras

Pada implementasi perangkat keras dilakukan pemasangan komponen-komponen perangkat keras pada PCB *board matrix* yang dapat dilihat pada Gambar 5.17 dan Gambar 5.18.



**Gambar 5.17 Implementasi hubungan NodeMCU dengan sensor dan aktuator**



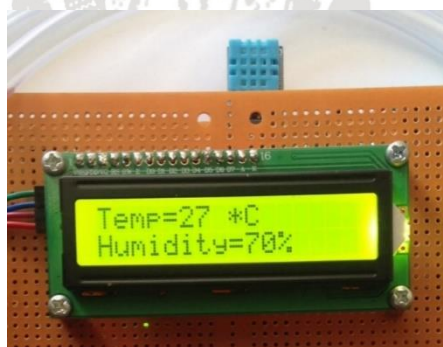
**Gambar 5.18 Implementasi *prototype* antarmuka**

Pada Gambar 5.17 dan 5.18 menampilkan hasil rancangan sistem dengan menggunakan 2 *PCB board* sebagai tempat untuk merangkai komponen agar tersusun dengan rapih. *PCB board* pada Gambar 5.12 menunjukkan beberapa komponen yang sudah terhubung dengan kabel *jumper* pada mikrokontroler NodeMCU seperti sensor DHT11, single relay, LM2596 DC, konektor klip, baterai dan pompa air, akan tetapi pada gambar tersebut pompa air tidak masuk dalam *frame*.

Kemudian pada Gambar 5.18 pemasangan layar LCD pada *PCB board* dipasang merangkap dengan *PCB board* sebelumnya dengan menggunakan *spacer* yang dipasang mur baut di setiap sisi PCB agar antar komponen tidak tertekan. LCD ditaruh di bagian luar sebagai antarmuka pengguna.

#### 5.2.1.1 Implementasi Subsistem Sensor DHT11

Pada implementasi subsistem sensor DHT11 diletakkan di antara papan pcb, implementasi dilakukan sesuai dengan perancangan subsistem sensor DHT11. Hasil implementasi dapat dilihat pada Gambar 5.19.



**Gambar 5.19 Implementasi subsistem sensor DHT11**

Pada Gambar 5.19 menampilkan hasil implementasi nilai sensor DHT11 yang sudah terhubung dengan LCD. Hasil nilai suhu dan kelembapan didapat dari masukan sensor yang diproses mikrokontroler. Data informasi ditampilkan pada layar LCD sebagai hasil pemantauan.

#### 5.2.1.2 Implementasi Subsistem Komunikasi NodeMCU dengan Blynk

Pada implementasi subsistem komunikasi NodeMCU dengan Blynk. Dilakukan implementasi sesuai dengan perancangan subsistem komunikasi NodeMCU dengan menggunakan modul *wi-fi* pada NodeMCU yaitu ESP8266 yang di

hubungkan dengan *smartphone* yang sudah terpasang aplikasi Blynk. Hasil implementasi dapat dilihat pada Gambar 5.20.



**Gambar 5.20 Implementasi subsistem komunikasi NodeMCU dengan Blynk**

Pada Gambar 5.20 menampilkan hasil implementasi pemantauan melalui aplikasi Blynk dan pemantauan pada LCD. Pada gambar tersebut menampilkan nilai informasi yang sama berarti pemantauan dapat dikatakan sudah sesuai. Hasil ini didapat dari pembacaan sensor yang diproses oleh NodeMCU dan pengguna dapat melihat hasil pemantauan pada aplikasi Blynk.

#### 5.2.1.3 Implementasi Subsistem Penyiraman

Pada implementasi subsistem penyiraman eksekusi dapat dilakukan ketika suhu dan kelembapan mengalami perubahan, perubahan ini akan diatur oleh *fuzzy* untuk melakukan penyiraman atau tidak. Penyiraman akan dilakukan ketika mikrokontroler mengirimkan masukan ke Relay untuk melakukan *switch on* agar pompa menyala. Hasil implementasi subsistem penyiraman dapat dilihat pada Gambar 5.21.



**Gambar 5.21 Implementasi subsistem penyiraman**

Pada Gambar 5.21 merupakan implementasi penyiraman yang dilakukan pada kubung jamur, penyiraman dilakukan dengan menunggu masukan sensor DHT11 yang dikirim ke mikrokontroler NodeMCU, hasil nilai suhu dan kelembapan yang sudah didapat akan di proses NodeMCU dan proses logika *fuzzy*, dengan melalui 3 tahap yaitu fuzzifikasi, inferensi dan defuzzifikasi. Kemudian hasil *output* tersebut didapat *rule fuzzy* yang berfungsi untuk kendali penyiraman maupun tidak menyiram.

#### 5.2.1.4 Implementasi Komponen Keseluruhan

Pada implementasi komponen keseluruhan yang dilakukan pada tempat kubung jamur, Balai Pengkajian Teknologi Pertanian, Jawa Timur. Implementasi dilakukan agar mengetahui perbedaan pada saat melakukan pemantauan dan penyiraman yang dilakukan oleh petani jamur.

Dapat dilihat pada Gambar 5.22 menampilkan penempatan perangkat sistem dalam kubung jamur pada Balai Pengkajian Teknologi Pertanian. Ukuran kubung jamur 3m x 7m sehingga dalam kubung terdapat 4 rak. Hasil data sensor dapat dipantau dengan mengunjungi kubung jamur untuk melihat hasil data pada layar LCD dan apabila tidak sempat, pemantauan juga bisa dilakukan dari jarak jauh melalui aplikasi Blynk.



Perangkat sistem

Gambar 5.22 Implementasi subsistem komponen keseluruhan

#### 5.2.2 Implementasi Perangkat Lunak

Pada implementasi perangkat lunak berisi penulisan sumber kode program yang berfungsi sebagai instruksi sistem pada perangkat keras. Pada Tabel 5.6 menampilkan program autentikasi token aplikasi Blynk yang terhubung pada modul *wi-fi* ESP8266 NodeMCU melalui perantara akses poin.

Tabel 5.6 Kode Program autentikasi token Blynk

Baris	Kode Program
1	<code>char auth[] = "3ca9782e5d6b40cc9d513189f119f54e";</code>
2	<code>char ssid[] = "iPhone";</code>
3	<code>char pass[] = "kosan43b";</code>

Pada Tabel 5.6 baris ke-1 menggunakan tipe data `char` dengan fungsi *auth* untuk mengaktifkan autentikasi Blynk token yang dikirim melalui email pada saat mendaftar aplikasi Blynk, lalu Blynk mengirim kode token berupa `"3ca9782e5d6b40cc9d513189f119f54e"` untuk diterapkan pada kode program setelah itu kode tersebut dimasukan ke dalam program Arduino IDE. Kemudian dilanjutkan pada baris ke-2 dengan memasukkan *ssid* dengan nama `"iPhone"` dan baris ke-3 memasukkan *password* `"kosan43b"`. Ketika sudah melakukan konfirmasi Blynk dapat terhubung dengan sistem.



**Tabel 5.7 Kode Program *setup***

Baris	Kode Program
1	void setup()
2	{
3	Serial.begin(9600);
4	pinMode(D5, OUTPUT);
5	Wire.begin(D3, D4);
6	lcd.begin(16, 2);
7	lcd.backlight();
8	Blynk.begin(auth, ssid, pass);
9	}

Pada Tabel 5.7 merupakan *setup program* yang digunakan untuk menginisialisasi variabel, pin dan proses lain yang digunakan dalam sketsa program. Baris ke-2 berfungsi sebagai pembuka program variabel *setup*. Baris ke-3 digunakan untuk memproses serial monitor pada Arduino IDE dengan menggunakan baudrate sebesar 9600. Baris ke-4 digunakan untuk memproses pin mode dengan data pin *D5* sebagai *output*, *output* dari NodeMCU ini dikirimkan ke relay sebagai *input* untuk melakukan *switch on/off*. Baris ke-5 sampai ke-7 menginisialisasi variabel LCD dengan memasukkan pin *SCL* ke pin *D4* dan pin *SDA* ke pin *D3*, kode *lcd.begin* (16,2) menandakan penggunaan LCD 16x2 dan *lcd.backlight* untuk menyalakan LED pada layar LCD. Baris ke-8 berisi konfigurasi Blynk seperti pada penjelasan Tabel 5.5 yaitu untuk mencocokkan antara konfigurasi Blynk dengan sistem program. Baris ke-9 berfungsi sebagai penutup program variabel *setup*.

**Tabel 5.8 Kode Program *loop***

Baris	Kode Program
1	void loop()
2	{
3	Blynk.run();
4	float x=temperature;
5	float y=humidity;
6	}
7	Serial.print("Sample OK: ");
8	Serial.print((int)temperature); Serial.print(" *C, ");
9	Serial.print((int)humidity); Serial.println(" H");
10	lcd.setCursor(0,0);
11	lcd.print("Temp=");
12	lcd.print((int)temperature);
13	lcd.print(" *C");
14	lcd.setCursor(0,1);
15	lcd.print("Humidity=");
16	lcd.print((int)humidity);
17	lcd.print("% ");

Pada Tabel 5.8 merupakan *loop program* untuk menjalankan suatu siklus program, yang akan dilakukan secara terus-menerus hingga NodeMCU mati/*reset*. Baris ke-2 perintah pembuka program variabel *loop*. Baris ke-3 berisi program perintah untuk mengaktifkan perintah Blynk. Baris ke-4 menggunakan tipe data float dengan menampilkan pemanggilan variabel yang terdapat pada fuzzifikasi dengan variabel "x" sebagai inisialisasi *temperature*, baris ke-5 juga menggunakan tipe data dengan variabel "y" sebagai inisialisasi *humidity*. Baris ke-6 perintah penutup program variabel *loop*. Baris ke-7 menuliskan sample ok pada serial monitor. Baris ke-8 menuliskan hasil data suhu pada serial monitor dengan diakhiri

penulisan “\*C” setelah hasil data suhu. Baris ke-9 menuliskan hasil data kelembapan pada serial monitor dengan diakhiri penulisan “%” setelah hasil data kelembapan. Baris ke-10 berfungsi sebagai set cursor untuk menampilkan data pada baris pertama layar LCD. Baris ke-11 menuliskan “Temp=” pada layar LCD. Baris ke-12 menuliskan hasil data suhu. Baris ke-13 menuliskan “\*C” setelah hasil data suhu. Baris ke-14 berfungsi sebagai set cursor untuk menampilkan data pada baris kedua layar LCD. Baris ke-15 menuliskan “Humidity=” pada layar LCD. Baris ke-16 menuliskan hasil data kelembapan. Baris ke-17 menuliskan “%” setelah hasil data kelembapan.

### 5.2.2.1 Fuzzifikasi

Implementasi fuzzifikasi dilakukan berdasarkan hasil data sensor dengan mengubah nilai sensor menjadi himpunan keanggotaan *fuzzy*. Kode program untuk fuzzifikasi berupa keanggotaan sensor suhu dan sensor kelembapan seperti pada Tabel 5.9 menampilkan program fuzzifikasi.

**Tabel 5.9 Kode Program fuzzifikasi**

Baris	Kode Program
1	float sensorsuhu(float x) {
2	Serial.print("x suhu = ");
3	Serial.println(x);
4	
5	if (x <= 18) {
6	ss[0] = 1;
7	} else if (x > 18 && x <= 22) {
8	ss[0] = ((22 - x) / (22 - 18));
9	} else if (x > 22){
10	ss[0] = 0;
11	}
12	
13	if (x <= 18) {
14	ss[1] = 0;
15	} else if (x > 18 && x <= 22) {
16	ss[1] = ((x - 18) / (22 - 18));
17	} else if (x > 22 && x <= 26) {
18	ss[1] = 1;
19	} else if (x >= 30){
20	ss[1] = 0;
21	}
22	
23	if (x <= 26) {
24	ss[2] = 0;
25	} else if (x > 26 && x <= 30) {
26	ss[2] = ((x - 26) / (30 - 26));
27	} else if (x >= 30) {
28	ss[2] = 1;
29	}
30	}
31	float sensorkelembaban(float y) {
32	Serial.print("y kelembaban = ");
33	Serial.println(y);
34	
35	if (y <= 75) {
36	sk[0] = 1;
37	} else if (y > 75 && y <= 80) {
38	sk[0] = ((80 - y) / (80 - 75));
39	} else if (x > 80){
40	sk[0] = 0;

```

41     }
42
43     if (y <= 75) {
44         sk[1] = 0;
45     } else if (y <= 75 && y >= 80) {
46         sk[1] = ((x - 75) / (80 - 75));
47     } else if (y > 80 && y <= 90) {
48         sk[1] = 1;
49     } else if (x >= 95) {
50         sk[1] = 0;
51     }
52
53     if (y <= 90) {
54         sk[2] = 0;
55     } else if (y > 90 && y <= 95) {
56         sk[2] = ((y - 90) / (95 - 90));
57     } else if (x >= 95) {
58         sk[2] = 1;
59     }
60 }

```

Pada baris ke-1 menggunakan tipe data float dengan variabel sensor suhu dan parameter "float x". Baris ke-2 menuliskan "x suhu = ". Baris ke-3 menuliskan "x" diakhiri dengan kalimat selanjutnya ada di baris berikutnya. Himpunan keanggotaan sensor "dingin" baris 5-11 jika "x" kurang dari sama dengan 18 bernilai 1, maka jika "x" lebih dari 18 dan "x" kurang dari sama dengan 22 bisa bernilai 0 dan 1, maka jika "x" lebih dari 22 bernilai 0. Keanggotaan sensor "sedang" baris 13-21 jika "x" kurang dari sama dengan 18 bernilai 0, maka jika "x" lebih dari 18 dan "x" kurang dari sama dengan 22 bisa bernilai 0 dan 1, maka jika "x" lebih dari 22 dan "x" kurang dari sama dengan 26 bernilai 1, maka jika "x" lebih dari sama dengan 30 bernilai 0. Keanggotaan sensor "panas" baris 23-29 jika "x" kurang dari sama dengan 26 maka bernilai 0, maka jika "x" lebih dari 26 dan "x" kurang dari sama dengan 30 bisa bernilai 0 dan 1, maka jika "x" lebih dari sama dengan 30 bernilai 1.

Baris ke-31 menggunakan tipe data float dengan variabel sensor kelembapan dan parameter "float y". Baris ke-32 menuliskan "y kelembapan = ". Baris ke-33 menuliskan "y" diakhiri dengan kalimat selanjutnya ada di baris berikutnya. Himpunan keanggotaan sensor "kering" baris 35-41 jika "y" kurang dari sama dengan 75 maka bernilai 1, maka jika "y" lebih dari 75 dan "y" kurang dari sama dengan 80 bisa bernilai 0 dan 1, maka jika "y" lebih dari 80 bernilai 0. Keanggotaan sensor "normal" baris 43-51 jika "y" kurang dari sama dengan 75 maka bernilai 0, maka jika "y" kurang dari sama dengan 75 dan "y" lebih dari sama dengan 80 bisa bernilai 0 dan 1, maka jika "y" lebih dari 80 dan "y" kurang dari 90 bernilai 1, maka jika "y" lebih dari sama dengan 95 bernilai 0. Keanggotaan sensor "basah" baris 53-59 jika "y" kurang dari sama dengan 90 bernilai 1, maka jika "y" lebih dari 90 dan "y" kurang dari sama dengan 95 bisa bernilai 0 dan 1, maka jika "y" lebih dari sama dengan 95 bernilai 1.

Program fuzzifikasi dengan menggunakan percabangan sebuah kondisi *else-if*, jika kondisi yang sedang diuji sudah terpenuhi, maka program akan menjalankan perintah yang sudah ditentukan. Jika kondisi yang diuji berbeda atau salah, maka program akan menjalankan perintah lain.

### 5.2.2.2 Inferensi

Inferensi dilakukan setelah melakukan proses fuzzifikasi yaitu himpunan keanggotaan sensor. Hasil dari keanggotaan sensor dibagi menjadi beberapa *rule fuzzy*. Kode program untuk inferensi dapat dilihat pada Tabel 5.10.

**Tabel 5.10 Kode Program inferensi**

Baris	Kode Program
1	void infer() {
2	int h = 0;
3	for (int i = 0; i < 3; i++) {
4	ss[i];
5	for (int j = 0; j < 3; j++) {
6	sk[j];
7	rules[h] = min(ss[i], sk[j]);
8	h += 1;
9	}
10	}
11	}

Pada Tabel 5.10 baris ke-1 menjelaskan mengenai program inferensi berisi mengenai proses yang berasal dari kesimpulan pada *premis-premis* program fuzzifikasi yang dianggap benar. Baris ke-2 menggunakan tipe data integer dengan variabel "h" sama dengan 0. Baris ke-3 menggunakan *looping for* dengan menggunakan *pre-increment* sebagai sebuah ekspresi sensor suhu. Baris ke-4 perulangan hasil evaluasi dari *post-increment* sensor suhu. Baris ke-5 menggunakan *looping for* dengan menggunakan *post-increment* sebagai sebuah ekspresi sensor kelembapan. Baris ke-6 perulangan hasil evaluasi dari *post-increment* sensor kelembapan. Baris ke-7 digunakan untuk mengevaluasi nilai suhu dan kelembapan secara keseluruhan hingga mendapatkan kumpulan *rule* yang selanjutnya akan diteruskan pada proses defuzzifikasi.

### 5.2.2.3 Defuzzifikasi

Defuzzifikasi dilakukan setelah melakukan proses inferensi yaitu menentukan *rule-rule fuzzy*. *Rule fuzzy* yang didapat akan dieksekusi sebagai proses penentuan *output*. Kode program defuzzifikasi dapat dilihat pada Tabel 5.11.

**Tabel 5.11 Kode Program defuzzifikasi**

Baris	Kode Program
1	void defuzzyfikasi()
2	{
3	const byte size = 9;
4	float rawArray[size] = {rules[0], rules[1], rules[2],
5	rules[3], rules[4], rules[5],
6	rules[6], rules[7], rules[8]}
7	};
8	Array<float> array = Array<float>(rawArray, size);
9	defu_final = array.getMax();
10	}

Pada Tabel 5.11 baris ke-1 menjelaskan mengenai program defuzzifikasi yaitu *rule-rule*. pada program ini data didapat dari proses fuzzifikasi, *inferensi* dan dapat dilakukan defuzzifikasi untuk mendapatkan *output rule fuzzy*. Baris ke-2 perintah pembuka program variabel defuzzifikasi. Baris ke-3 proses defuzzifikasi terdapat 9 *rule* yang diperoleh dari hasil inferensi. Baris 4-7 menggunakan tipe data float



dengan variabel *rawArray* menggunakan parameter “size” menampilkan rules[0] sampai rules[8]. Baris ke-8 untuk menentukan parameter *rule fuzzy* yang akan di pilih. Baris ke-9 menentukan hasil akhir defuzzifikasi. Baris ke-10 perintah penutup program variabel defuzzifikasi. Proses defuzzifikasi akan mengirimkan *rule* yang sudah didapat dan kemudian dikirimkan ke program penyiraman untuk melakukan proses eksekusi.

#### 5.2.2.4 Penyiraman

Penyiraman kode program berisi perintah untuk menuliskan aturan secara digital pada suatu pin. Nilai keluaran dapat berupa *high* atau *low* pada suatu pin yang sudah disetel. Pada Tabel 5.12 menampilkan perintah program penyiraman.

**Tabel 5.12 Kode Program penyiraman**

Baris	Kode Program
1	if (defu_final == rules[0]) {
2	Serial.println("Rules 0");
3	digitalWrite(D5, HIGH);
4	delay(7500);
5	digitalWrite(D5, LOW);
6	delay(1000);
7	}
8	else if (defu_final == rules[1]) {
9	Serial.println("Rules 1");
10	digitalWrite(D5, LOW);
11	delay(2000);
12	}
13	else if (defu_final == rules[2]) {
14	Serial.println("Rules 2");
15	digitalWrite(D5, LOW);
16	delay(2000);
17	}
18	else if (defu_final == rules[3]) {
19	Serial.println("Rules 3");
20	digitalWrite(D5, HIGH);
21	delay(7500);
22	digitalWrite(D5, LOW);
23	delay(1000);
24	}
25	else if (defu_final == rules[4]) {
26	Serial.println("Rules 4");
27	digitalWrite(D5, LOW);
28	delay(2000);
29	}
30	else if (defu_final == rules[5]) {
31	Serial.println("Rules 5");
32	digitalWrite(D5, LOW);
33	delay(2000);
34	}
35	else if (defu_final == rules[6]) {
36	Serial.println("Rules 6");
37	digitalWrite(D5, HIGH);
38	delay(15000);
39	digitalWrite(D5, LOW);
40	delay(1000);
41	}
42	else if (defu_final == rules[7]) {
43	Serial.println("Rules 7");
44	digitalWrite(D5, HIGH);
45	delay(7500);
46	digitalWrite(D5, LOW);

47	delay(1000);
48	}
49	else if (defu_final == rules[8]) {
50	Serial.println("Rules 8");
51	digitalWrite(D5, LOW);
52	delay(2000);
53	}

Pada Tabel 5.12 baris ke-1 jika hasil *defu\_final* sama dengan rules 0. Baris ke-2 menuliskan "rules 0". Baris ke-3 menuliskan nilai secara digital pada pin *D5* berupa keluaran "HIGH". Baris ke-4 perintah *delay* selama 7,5 detik. Baris ke-5 menuliskan nilai secara digital pada pin *D5* berupa keluaran "LOW". Baris ke-6 perintah *delay* selama 1 detik. Baris ke-8 maka jika hasil *defu\_final* sama dengan rules 1. Baris ke-9 menuliskan "rules 1". Baris ke-10 menuliskan nilai secara digital pada pin *D5* berupa keluaran "LOW". Baris ke-11 perintah *delay* selama 2 detik. Baris ke-13 maka jika hasil *defu\_final* sama dengan rules 2. Baris ke-14 menuliskan "rules 2". Baris ke-15 menuliskan nilai secara digital pada pin *D5* berupa keluaran "LOW". Baris ke-16 perintah *delay* selama 2 detik. Baris ke-18 maka jika hasil *defu\_final* sama dengan rules 3. Baris ke-19 menuliskan "rules 3". Baris ke-20 menuliskan nilai secara digital pada pin *D5* berupa keluaran "HIGH". Baris ke-21 perintah *delay* selama 7,5 detik. Baris ke-22 menuliskan nilai secara digital pada pin *D5* berupa keluaran "LOW". Baris ke-23 perintah *delay* selama 2 detik. Baris ke-25 maka jika hasil *defu\_final* sama dengan rules 4. Baris ke-26 menuliskan "rules 4". Baris ke-27 menuliskan nilai secara digital pada pin *D5* berupa keluaran "LOW". Baris ke-28 perintah *delay* selama 2 detik. Baris ke-30 maka jika hasil *defu\_final* sama dengan rules 5. Baris ke-31 menuliskan "rules 5". Baris ke-32 menuliskan nilai secara digital pada pin *D5* berupa keluaran "LOW". Baris ke-33 perintah *delay* selama 2 detik. Baris ke-35 maka jika hasil *defu\_final* sama dengan rules 6. Baris ke-36 menuliskan "rules 6". Baris ke-37 menuliskan nilai secara digital pada pin *D5* berupa keluaran "HIGH". Baris ke-38 perintah *delay* selama 15 detik. Baris ke-39 menuliskan nilai secara digital pada pin *D5* berupa keluaran "LOW". Baris ke-40 perintah *delay* selama 1 detik. Baris ke-42 maka jika hasil *defu\_final* sama dengan rules 7. Baris ke-43 menuliskan "rules 7". Baris ke-44 menuliskan nilai secara digital pada pin *D5* berupa keluaran "HIGH". Baris ke-45 perintah *delay* selama 15 detik. Baris ke-46 menuliskan nilai secara digital pada pin *D5* berupa keluaran "LOW". Baris ke-47 perintah *delay* selama 1 detik. Baris ke-49 maka jika hasil *defu\_final* sama dengan rules 9. Baris ke-50 menuliskan "rules 9". Baris ke-51 menuliskan nilai secara digital pada pin *D5* berupa keluaran "LOW". Baris ke-52 perintah *delay* selama 2 detik.

Program penyiraman menggunakan percabangan *if-else* jika kondisi yang sedang diuji belum terpenuhi, maka program akan melanjutkan ke kondisi yang lain sampai kondisi dapat terpenuhi. Kemudian hasil pada program penyiraman ini dijadikan sebagai acuan dalam melakukan penyiraman tanaman kubung jamur.

## BAB 6 PENGUJIAN DAN ANALISIS

Pengujian dan analisis membahas tahap pengujian sistem yang sebelumnya telah dirancang dan diimplementasi. Pengujian dilakukan untuk membandingkan hasil analisis pengujian sistem yang dikerjakan pada penelitian sebelumnya. Pengujian dan analisis dibagi menjadi 3 tahap yaitu pengujian sensor DHT11, pengujian pengamatan dan pengujian metode *fuzzy* untuk melakukan eksekusi *rule fuzzy* berupa penyiraman.

### 6.1 Pengujian Sensor DHT11

Pengujian pertama dengan melakukan pengujian terhadap akurasi pembacaan sensor DHT11. Pengujian dilakukan dengan melihat hasil pembacaan data yang diperoleh dari nilai suhu dan kelembapan dengan hasil pada alat ukur termometer yang berada di kubung jamur.

#### 6.1.1 Tujuan Pengujian

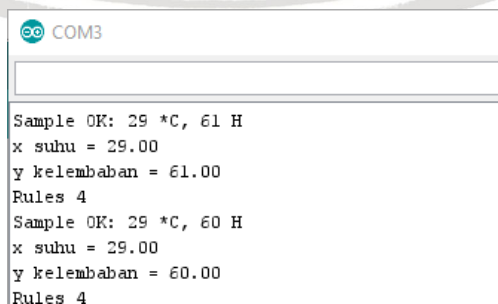
Tujuan dari pengujian sensor DHT11 adalah untuk mengetahui perbandingan akurasi dari pembacaan sensor DHT11 dengan alat ukur termometer. Dengan melihat hasil akurasi kedua sensor, maka hasil perbandingan dapat diketahui dengan menghitung analisis rata-rata *error* sensor.

#### 6.1.2 Prosedur Pengujian

Prosedur pengujian akurasi sensor DHT11 dilakukan dalam beberapa langkah sebagai berikut:

1. Menghubungkan sensor DHT11 dengan NodeMCU sesuai Gambar 5.2
2. Menghubungkan NodeMCU dengan laptop menggunakan kabel *Universal Serial Bus* (USB).
3. Menjalankan kode program sistem pemantauan yang ada pada NodeMCU
4. Mengamati data hasil pembacaan sensor DHT11 suhu dan kelembapan yang ditampilkan pada *serial monitor* untuk mengetahui tingkat akurasi kesalahan dan keberhasilan.

#### 6.1.3 Hasil Pengujian



```
COM3
Sample OK: 29 *C, 61 H
x suhu = 29.00
y kelembapan = 61.00
Rules 4
Sample OK: 29 *C, 60 H
x suhu = 29.00
y kelembapan = 60.00
Rules 4
```

Gambar 6.1 Hasil pengujian sensor

Gambar 6.1 menampilkan hasil *rules fuzzy* dan juga menampilkan hasil nilai sensor DHT11 yaitu suhu dan kelembapan yang sudah diproses oleh mikrokontroler NodeMCU, hasil tersebut ditampilkan melalui *serial monitor* pada Arduino IDE.

#### 6.1.4 Analisis Pengujian

Adapun analisis pengujian sensor DHT11 dengan alat ukur termometer, sensor DHT11 memiliki keunggulan, karena mempunyai 2 keluaran sekaligus yaitu suhu dan kelembapan, sedangkan termometer hanya mengukur suhu. Adapun kekurangan dari sensor DHT11, sensor harus dihubungkan terlebih dahulu dengan mikrokontroler yang sudah ditulis program agar bisa mendeteksi nilai suhu dan kelembapan. Perbandingan hasil pengujian kedua sensor yaitu sebagai berikut:

**Tabel 6.1 Analisis perbandingan pengujian sensor**

No	Suhu DHT11	Suhu Termometer	Error %
1	21	22	4,54
2	27	28	3,57
3	26	27	3,70
4	23	22	4,54
5	24	26	7,69
6	26	27	3,70
7	23	22	4,54
8	25	27	7,40
9	21	22	4,54
10	22	22	0
11	24	26	7,69
12	25	27	7,40
13	21	20	5
14	23	25	8
15	24	25	4
16	22	21	4,76
17	24	25	4
18	25	25	0
19	23	23	0
20	20	22	9,09
21	25	26	3,84



22	21	21	0
23	22	23	4,34
24	24	24	0
25	22	23	4,34
26	25	26	3,84
27	24	25	4
28	24	24	0
29	27	28	3,57
30	25	24	4,16
Rata-rata			4,07

Dari data yang diperoleh pada Tabel 6.1 pengujian didapatkan bahwa hasil dari pembacaan sensor DHT11 yang dilakukan memiliki rata-rata persentase *error* sebesar 4,07 %. Contoh perhitungan manual persentase *error* sebagai berikut:

Nilai pembacaan sensor = 21

Nilai pembacaan termometer = 22

$$\begin{aligned}
 \text{Selisih nilai pembacaan} &= |\text{pembacaan alat ukur} - \text{pembacaan sensor}| \\
 &= |22 - 21| \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
 \text{Persentase error} &= \frac{\text{Selisih nilai pembacaan}}{\text{Pembacaan alat ukur}} \times 100\% \\
 &= \frac{1}{22} \times 100\% \\
 &= 4,54\%
 \end{aligned}$$

Adapun cara untuk menghitung nilai rata-rata *error* keseluruhan pengujian adalah sebagai berikut:

$$\begin{aligned}
 \text{Rata-rata error} &= \frac{\text{Jumlah persentase error}}{\text{Jumlah pengujian}} \\
 &= \frac{122,25}{30} \\
 &= 4,07
 \end{aligned}$$

Berdasarkan hasil perhitungan didapatkan nilai rata-rata *error* secara keseluruhan yakni sebesar 4,07% yang mana hasil tersebut didapat dari hasil perbandingan antara nilai suhu pada sensor dan nilai suhu pada termometer. Akurasi pembacaan sensor DHT11 dapat dikatakan bahwa sensor sudah sesuai dengan kondisi ruang kubung jamur. Selain itu, meskipun masih terdapat *error* antara pembacaan sensor dengan termometer, namun sensor tetap dapat membaca nilai suhu dan kelembapan meskipun perubahan nilainya relatif kecil.

## 6.2 Pengujian Pengamatan

Pengujian kedua dengan melakukan pembacaan sensor DHT11 agar dapat menampilkan nilai suhu dan kelembapan. Pengujian dilakukan dengan membandingkan hasil yang didapat dari alat ukur termometer dengan hasil yang didapat dari data sensor melalui layar LCD dan aplikasi Blynk.

### 6.2.1 Tujuan Pengujian

Tujuan dari pengujian pengamatan untuk membandingkan hasil pemantauan yang dilakukan petani tanaman jamur dengan sistem yang dikerjakan penulis. Hasil sistem pemantauan yang dikerjakan penulis dapat dilihat pada layar LCD dan juga dapat di pantau dari jauh menggunakan aplikasi Blynk.

### 6.2.2 Prosedur Pengujian

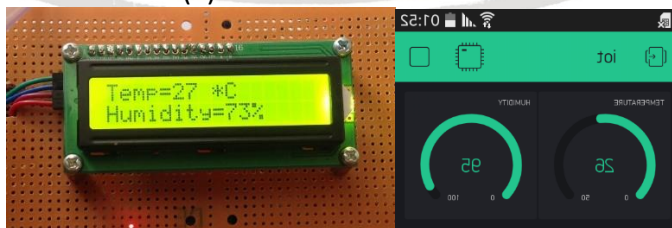
Prosedur pengujian dilakukan dalam beberapa tahap sebagai berikut:

1. Menghubungkan komponen LCD dengan NodeMCU sesuai Gambar 5.3
2. Menghubungkan NodeMCU dengan laptop menggunakan kabel USB
3. Menjalankan kode program komponen LCD dan Blynk yang ada pada NodeMCU
4. Menampilkan data hasil sensor pada LCD dan aplikasi Blynk
5. Menghubungkan *smartphone* dengan *ssid* dan *password* akses poin yang sudah di tentukan program pada Tabel 5.5

### 6.2.3 Hasil Pengujian



(a) alat ukur termometer



(b) LCD

(c) Blynk

**Gambar 6.2 Hasil perbandingan pengujian alat ukur**

Pada Gambar 6.2 menampilkan hasil perbandingan pengujian pemantauan data suhu dan kelembapan. Pada bagian (a) petani jamur menggunakan alat ukur

berupa termometer untuk mengetahui kondisi suhu pada kubung jamur. Sedangkan sistem yang dibuat pemantauan dapat dilihat pada layar (b) dengan menggunakan LCD sebagai antarmuka untuk melihat data hasil sensor dan (c) dengan menggunakan Blynk sebagai antarmuka pengguna untuk melihat data hasil sensor. Perangkat aplikasi Blynk memungkinkan pengguna dapat memantau suhu dan kelembapan dari jauh. Hasil pengujian pada ESP8266 NodeMCU dan konektivitas Blynk dengan koneksi internet didapatkan ping Blynk yaitu 4ms.

#### 6.2.4 Analisis Pengujian

Pada bagian analisis pengujian sistem pemantauan yang dikerjakan peneliti terdapat 2 bagian yaitu dengan memantau hasil data sensor pada layar LCD dan aplikasi Blynk. Pengujian sistem memiliki keunggulan dalam pemantauan suhu dan kelembapan dengan melihat hasil data pada layar LCD dan menggunakan perangkat *smartphone* sebagai antarmuka aplikasi Blynk untuk pemantauan dari jauh. Sedangkan pemantauan yang dilakukan petani hanya melihat pada alat ukur termometer. Adapun kekurangan dari pengujian sistem pemantauan yaitu terdapat delay setiap perubahan kondisi suhu dan kelembapan yang menjadikan pengiriman dan penerimaan data menjadi lambat.

#### 6.3 Pengujian Metode Logika Fuzzy

Pada pengujian dilakukan dengan menerapkan penyiraman dengan metode logika *fuzzy* sebagai kendali penyiraman otomatis. Penyiraman dilakukan dengan membandingkan hasil pengujian penyiraman metode logika *fuzzy* dengan penyiraman manual yang dilakukan oleh petani.

##### 6.3.1 Tujuan Pengujian

Tujuan pengujian metode *fuzzy* untuk mengetahui hasil perbandingan penyiraman menggunakan metode logika *fuzzy* dengan penyiraman manual, sehingga dapat diketahui tingkat perbedaan durasi dan banyaknya air yang dikeluarkan.

##### 6.3.2 Prosedur Pengujian

Prosedur pengujian dilakukan dalam beberapa tahap sebagai berikut:

1. Menghubungkan komponen relay, step down 2596, baterai dan pompa air dengan NodeMCU sesuai Gambar 5.4
2. Menghubungkan NodeMCU dengan laptop menggunakan kabel USB
3. Menjalankan kode program penyiraman dengan metode logika *fuzzy* pada NodeMCU
4. Mengamati hasil penyiraman metode logika *fuzzy* dengan penyiraman manual. Kemudian mencatat perbedaan durasi dan banyaknya air sebagai perbandingan data hasil pengujian

### 6.3.3 Hasil Pengujian

Tabel 6.2 Hasil pengujian penyiraman

Uji ke-	Waktu	Suhu	Kelembapan	Rule	Penyiraman	Keterangan
1	08.14	21	88	Rule 1	Off	Berhasil
2	12.20	27	78	Rule 6	Banyak	Berhasil
3	15.18	26	79	Rule 3	Sedikit	Berhasil
4	08.02	23	92	Rule 5	Off	Berhasil
5	11.17	24	79	Rule 3	Sedikit	Berhasil
6	15.33	26	81	Rule 4	Off	Berhasil
7	08.11	23	95	Rule 5	Off	Berhasil
8	12.47	25	78	Rule 3	Sedikit	Berhasil
9	15.37	21	86	Rule 1	Off	Berhasil
10	08.23	22	92	Rule 5	Off	Berhasil
11	12.42	24	73	Rule 3	Sedikit	Berhasil
12	15.32	25	71	Rule 3	Sedikit	Berhasil
13	08.02	21	94	Rule 2	Off	Berhasil
14	12.23	23	72	Rule 3	Sedikit	Berhasil
15	15.33	24	80	Rule 4	Off	Berhasil
16	08.17	22	95	Rule 5	Off	Berhasil
17	12.43	24	78	Rule 3	Sedikit	Berhasil
18	15.30	25	89	Rule 4	Off	Berhasil
19	08.25	23	90	Rule 5	Off	Berhasil
20	12.35	20	76	Rule 0	Sedikit	Berhasil
21	15.32	25	84	Rule 4	Off	Berhasil
22	08.13	21	94	Rule 2	Off	Berhasil
23	12.13	22	77	Rule 3	Sedikit	Berhasil
24	15.37	24	86	Rule 4	Off	Berhasil
25	09.19	22	80	Rule 4	Off	Berhasil
26	12.18	25	76	Rule 3	Sedikit	Berhasil
27	15.28	24	79	Rule 3	Sedikit	Berhasil
28	08.30	24	93	Rule 5	Off	Berhasil



29	13.20	27	80	Rule 7	Sedikit	Berhasil
30	15.50	25	93	Rule 5	Off	Berhasil

Pada Tabel 6.2 menampilkan pengujian penyiraman pada kubung jamur untuk menyesuaikan apakah penyiraman sudah sesuai dengan *rule fuzzy* yang sudah ditentukan dengan akurasi 100%. Jam penyiraman menyesuaikan dengan petani kubung jamur agar mengetahui berapa banyak air yang dikeluarkan dan berapa lama waktu untuk melakukan penyiraman. Adapun proses perhitungannya yaitu:

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Total data} - \text{Data tidak sesuai}}{\text{Total data}} \times 100 \\
 &= \frac{30 - 0}{30} \times 100 \\
 &= \frac{30}{30} \times 100 \\
 &= 100\%
 \end{aligned}$$

Dalam pengujian metode logika *fuzzy* model *Sugeno* dapat dikatakan berhasil ketika eksekusi penyiraman sesuai dengan metode *fuzzy*. Sistem akan dianggap gagal apabila eksekusi penyiraman tidak sesuai dengan *rule fuzzy* yang sudah ditentukan.

#### 6.3.4 Analisis Pengujian

Adapun hasil analisis perbandingan pengujian antara penyiraman metode logika *fuzzy* dengan penyiraman manual. Keunggulan dari penyiraman dengan metode logika *fuzzy*, penyiraman dilakukan secara otomatis sesuai dengan perubahan kondisi suhu dan kelembapan. Penyiraman manual dilakukan dengan melihat nilai suhu pada alat ukur termometer, apabila suhu lebih dari 26°C maka dilakukan penyiraman. Adapun kekurangan dari sistem penyiraman dengan metode *fuzzy*, tegangan pompa kurang besar sehingga kurangnya tekanan yang diberikan pompa ketika melakukan penyiraman. Dapat dilihat pada Tabel 6.3 menampilkan hasil perbandingan pengujian penyiraman.

**Tabel 6.3 Analisis perbandingan pengujian penyiraman**

Uji ke-	Jam	DHT11		Termometer	Penyiraman Fuzzy		Penyiraman Manual	
		Suhu	Kelembapan	Suhu	detik	ml	Detik	ml
1	08.14	21	88	22	0	0	0	0
2	09.01	23	85	24	0	0	0	0
3	10.09	25	82	26	0	0	0	0
4	11.03	26	79	27	7,5	250	0	0
5	12.20	27	78	28	15	500	120	3000
6	13.12	27	77	29	15	500	0	0

7	14.04	28	77	30	15	500	0	0
8	15.18	26	79	27	7,5	250	0	0
9	16.02	22	82	23	7,5	250	0	0
10	17.00	20	85	22	7,5	250	0	0
<b>Jumlah</b>					75	2500	120	3000

Pada Tabel 6.3 menampilkan hasil analisis perbandingan pengujian penyiraman *fuzzy* dengan penyiraman manual. Pengujian dilakukan dengan melihat perbandingan waktu kira-kira 1 jam dalam sehari, pengujian penyiraman *fuzzy* penyiraman akan dilakukan dengan menentukan sensor suhu dan kelembapan yang sudah dipasang di dalam kubung jamur, penyiraman dilakukan sesuai dengan aturan yang sudah ditentukan dengan *rule fuzzy*. Penyiraman menggunakan metode *fuzzy*, air yang dikeluarkan sebanyak 250 ml air dengan durasi 7,5 detik untuk penyiraman sedikit dan 500 ml air untuk penyiraman banyak dengan durasi selama 15 detik. *Output* dari jumlah durasi sebanyak 75 detik atau sekitar 1,25 menit kemudian banyaknya jumlah penyiraman sebesar 2500 ml atau 2,5 liter.

Sedangkan pada pengujian penyiraman manual diketahui pada pagi hari yaitu pukul 08.14 tidak dilakukan penyiraman karena suhu optimal yaitu 22°C. Pada siang hari pukul 12.20 dilakukan penyiraman karena suhu 28°C penyiraman dilakukan sebanyak 3 liter dengan durasi 120 detik. Sedangkan pada sore hari pukul 16.02 tidak dilakukan penyiraman karena suhu optimal yaitu 23°C. Jumlah durasi dari pengujian manual yaitu sebesar 120 detik atau sekitar 2 menit kemudian jumlah penyiraman sebanyak 3000 ml atau 3 liter.

$$\begin{aligned}
 \text{Selisih nilai pembacaan} &= |\text{pembacaan alat ukur} - \text{pembacaan sensor}| \\
 &= |3000 - 2500| \\
 &= 500
 \end{aligned}$$

$$\begin{aligned}
 \text{Persentase error} &= \frac{\text{Selisih nilai pembacaan}}{\text{Pembacaan alat ukur}} \times 100\% \\
 &= \frac{500}{3000} \times 100\% \\
 &= 16,66\%
 \end{aligned}$$

Dari perbandingan penyiraman manual kurang sesuai karena hanya melihat kondisi suhu ruangan tanpa mengetahui kondisi kelembapan ruangan. Perbandingan durasi pengujian memiliki perbedaan 45 detik, sedangkan pada jumlah pengujian penyiraman memiliki perbedaan 500 ml atau 0,5 liter air. Persentase *error* perbandingan pengujian penyiraman sebesar 16,66%.

## BAB 7 PENUTUP

Pada bab ini memuat penarikan kesimpulan dari rumusan masalah yang didapat dari permasalahan penelitian sebelumnya, sesuai dengan tahapan-tahapan yang telah dikerjakan. Selain itu pada bab ini penulis menyampaikan saran yang diharapkan dapat digunakan dalam melakukan pengembangan selanjutnya.

### 7.1 Kesimpulan

Berdasarkan hasil dari tahap perancangan, implementasi, pengujian serta analisis hasil pengujian yang telah dilakukan, penulis dapat menarik kesimpulan bahwa:

1. Perangkat sistem pemantauan dan pengendalian penyiraman dimanfaatkan pada tanaman jamur sebagai alat bantu pemantauan ruang kubung jamur dan otomatisasi penyiraman tanaman jamur. Pemantauan merupakan suatu bentuk pengamatan data hasil suhu dan kelembapan yang didapat dari sensor, hasil pemantauan data sensor dapat dilihat pada antarmuka layar LCD dan aplikasi Blynk. Otomatisasi penyiraman tanaman jamur memanfaatkan metode logika *fuzzy* sebagai fungsi kendali penyiraman.
2. Implementasi metode logika *fuzzy* dapat dilakukan pada perangkat sistem pemantauan dan pengendalian penyiraman yang terdiri dari sensor suhu dan kelembapan DHT11, modul *step down* LM2596 dan aktuator Relay dengan memanfaatkan mikrokontroler NodeMCU yang diprogram pada lingkungan pengembangan Arduino. Perangkat sistem diimplementasikan pada ruang kubung jamur dengan metode *fuzzy Sugeno* yang dibagi menjadi 3 tahap yaitu fuzzifikasi, inferensi dan defuzzifikasi. Setelah itu didapatkan *rule fuzzy* yang akan eksekusi antara menyiram atau tidak menyiram.
3. Kinerja pengujian yang bertujuan untuk mengukur data hasil perbandingan perangkat sensor DHT11 dengan alat ukur termometer didapatkan rata-rata *error* 4,07%. Kinerja pengujian sistem dibandingkan dengan pemantauan pada alat ukur termometer, dengan melihat data hasil sensor pada layar LCD dan pemantauan dari jauh menggunakan aplikasi Blynk. Kinerja pengujian penyiraman manual dibandingkan dengan metode logika *fuzzy* model Sugeno didapatkan persentase *error* sebanyak 16,66%.

### 7.2 Saran

Berdasarkan kesimpulan yang telah didapatkan, terdapat beberapa saran untuk penelitian selanjutnya sebagai langkah pengembangan sistem ini, yaitu:

1. Menggunakan pompa dengan tegangan yang lebih besar sehingga *output* yang dikeluarkan lebih besar.

2. Membandingkan metode yang sudah digunakan pada penelitian ini dengan metode yang lain, untuk menentukan manakah metode yang memiliki akurasi lebih baik.
3. Memperbaiki antarmuka dan program *back-end* dari aplikasi android agar dapat membuat menu dan isi aplikasi yang sesuai dengan yang akan diimplementasikan.





## DAFTAR PUSTAKA

- Addicore, 2018. *Addicore*. [Online] Tersedia di: <<https://www.addicore.com/Single-Channel-Relay-Module-p/ad317.htm>> [Diakses 06 Juni 2018].
- Ajie, 2016. *Mengukur Suhu dan Kelembaban udara dengan DHT11 dan Arduino*. [Online] Tersedia di: <<http://saptaji.com/2016/08/10/mengukur-suhu-dan-kelembaban-udara-dengan-sensor-dht11-dan-arduino/>> [Diakses 31 Oktober 2018].
- Alat Labor, 2018. *Hygrometer*. [Online] Tersedia di: <<http://www.alatlabor.com/article/detail/95/manfaat-hygrometer-dan-jenis-jenis-hygrometer>> [Diakses 08 November 2018].
- AliExpress, 2018. *DC 12V Relay Module 1-channel 1 Channel High Level Trigger Relay Expansion Board*. [Online] Tersedia di: <<https://www.aliexpress.com/item/DC-12V-Relay-Module-1-channel-1-Channel-High-Level-Trigger-Relay-Expansion-Board/32320968386.html>> [Diakses 10 November 2018].
- Alifah, H., 2015. *satujam*. [Online] Tersedia di: <<http://www.satujam.com>> [Diakses 06 Februari 2018].
- Artofcircuit, 2018. *Artofcircuit*. [Online] Tersedia di: <<https://artofcircuits.com/product/dc-3-6v-120lh-mini-submersible-water-pump>> [Diakses 6 Juni 2018].
- Beavis, G., 2018. *Techradar*. [Online] Tersedia di: <<https://www.techradar.com>> [Diakses 24 Oktober 2018].
- Blynk, 2017. *Blynk*. [Online] Tersedia di: <<https://www.blynk.cc/>> [Diakses 06 February 2018].
- Cahaya, Y. A., Muchroddi & Bakrun, M., 1997. *Pembudidayaan dan Analisis Usaha Jamur Tiram*. Bogor, Penebar Swadaya.
- College Loan Consolidation, 2016. *Suhu dan Pengukurannya*. [Online] Tersedia di: <<http://fisikazone.com/suhu-dan-pengukurannya/>> [Diakses 08 November 2018].
- Components101, 2018. *DHT11–Temperature and Humidity Sensor*. [Online] Tersedia di: <<https://components101.com/dht11-temperature-sensor>> [Diakses 14 November 2018].
- Diymore, 2018. *1602 16X2 LCD IIC/I2C/TWI/SPI Serial Interface Module Display Blue/Yellow*. [Online] Tersedia di: <<https://www.diymore.cc/products/blue-1602-16x2-lcd-iic-i2c-twi-spi-serial-interface-module-display>> [Diakses 14 November 2018].

- electronics, 2018. *electronics*. [Online] Tersedia di: <electronics.stackexchange.com>
- ESP8266 Community Forum, 2018. *Arduino core for ESP8266*. [Online] Tersedia di: <<https://github.com/esp8266/Arduino>> [Diakses 13 Januari 2018].
- Espressif Systems, 2017. *ESP8266EX Datasheet*. [Online] Tersedia di: <[http://espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](http://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)> [Diakses 4 Februari 2018].
- Gunawan & Wydia, A., 2009. *Fungi Biotechnology*. 2 penyunt. Jakarta: Penerbit Atma Jaya.
- Immersa Lab, 2014. *Pengenalan Mikrokontroler*. [Online] Tersedia di: <<http://www.immersa-lab.com/pengenalan-mikrokontroler.htm>> [Diakses 06 November 2018].
- Istuti, W., 2009. *Balai Pengkajian Teknologi Pertanian Jawa Timur*. [Online] Tersedia di: <<http://jatim.litbang.pertanian.go.id>> [Diakses 05 February 2018].
- Itead, 2017. *LM2596 DC-DC Buck Converter Step-Down Power Module Output 1.25V-35V*. [Online] Tersedia di: <<https://www.itead.cc/lm2596-dc-dc-buck-converter-step-down-power-module-output-1-25v-35v.html>> [Diakses 09 November 2018].
- Jogja Robotika, 2018. *Mikrokontroler Atmega Murah*. [Online] Tersedia di: <<http://www.jogjarobotika.com/5-mikrokontroler>> [Diakses 08 November 2018].
- Juraj, 2018. *Arduino sketch for NodeMCU v1.0 (ESP8266-12E) has no error of compiling but does not run on NodeMCU v1.0 (ESP8266-12E) Board*. [Online] Tersedia di: <<https://arduino.stackexchange.com/posts/56093/revisions>> [Diakses 25 Oktober 2018].
- Kaewwiset, T. & Yodkhad, P., 2017. *Automatic Temperature and Humidity Control System by Using Fuzzy Logic*. Chiangrai, Thailand, Chiangrai College.
- Mikesdowdell, 2017. *ESP8266 NODEMCU V3 PINOUTS FOR MOSI/MOSO/CLK AND SDCARD*. [Online] Tersedia di: <<http://www.esp8266.com/viewtopic.php?f=13&t=15585>> [Diakses 7 February 2018].
- NodeMCU Team, 2014. *NodeMCU: An open-source firmware based on ESP8266*. [Online] Tersedia di: <[http://nodemcu.com/index\\_en.html](http://nodemcu.com/index_en.html)> [Diakses 4 Februari 2018].
- Nodemcu, 2014. *nodemcu*. [Online] Tersedia di: <<http://www.nodemcu.com>> [Diakses 06 February 2018].
- Nurbana, W. I. d. S., 2015. *Budidaya Jamur Tiram*. Malang, BPTP Jawa Timur.

- Orcott, M. G. H. a. D. M., 1987. *The Physiology of Plants Under Stress*. New York, John Willey and Sons.
- Parlindungan, A. K., 2000. *Pengaruh Konsentrasi Urea dan TSP di Dalam Air Rendaman Baglog Alang-alang terhadap Pertumbuhan dan Produksi Jamur Tiram Putih*. 2 penyunt. Pekanbaru: Dosen UNRI.
- Prathibha, S. R., Hongal, A. & Jyothi, M. P., 2017. *IOT Basen Monitoring Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*. Bangalore, India, IEEE.
- Prayitno, W. A., 2017. Sistem Monitoring Suhu, Kelembaban, dan Pengendali Penyiraman Tanaman Hidroponik menggunakan Blynk Android. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 1(2), pp. 292-297.
- Roger, P., 2006. *Mushroom*. 2 ed. Inggris: MacMillan.
- Sawyer, W. a., 2011. *Pengertian Smartphone Secara Umum*. [Online] Tersedia di: <<https://www.maxmanroe.com>> [Diakses 22 Oktober 2018].
- Seed Studio, 2017. *NodeMCU: Lua based ESP8266 development kit*. [Online] Tersedia di: <<https://www.seeedstudio.com/NodeMCU-v2-Lua-based-ESP8266-development-kit-p-2415.html>> [Diakses 11 November 2017].
- Stand Out Publishing, 2017. *Definition of Actuator*. [Online] Tersedia di: <<http://standoutpublishing.com/g/actuator.html>> [Diakses 08 November 2018].
- Suranata, A., 2017. *Tutorial Menggunakan LCD 2 Kabel pada I2C*. [Online] Tersedia di: <<https://tutorkeren.com/artikel/tutorial-menggunakan-lcd-dua-kabel-i2c-pada-nodemcu.htm>> [Diakses 31 Oktober 2018].
- Suriawiria, 1997. *Bioteknologi Perjamuran*. 1546922077388275712 penyunt. Yogyakarta: Angkasa.
- Terlaris.aseanpriceblog, 2018. *Harga Power Bank Veger 25000mAh V16 Original - Hitam Dan Spesifikasinya*. [Online] Tersedia di: <<https://terlaris.aseanpriceblog.org/harga-veger-power-bank-original-25000mah-hitam-dan-spesifikasinya/>> [Diakses 31 Oktober 2018].
- Tutorials Point, 2018. *Fuzzy Logic Inference System*. [Online] Tersedia di: <[https://www.tutorialspoint.com/fuzzy\\_logic/fuzzy\\_logic\\_inference\\_system.htm](https://www.tutorialspoint.com/fuzzy_logic/fuzzy_logic_inference_system.htm)> [Diakses 31 Oktober 2018].
- Tutorials Point, 2018. *Fuzzy Logic Introduction*. [Online] Tersedia di: <[https://www.tutorialspoint.com/fuzzy\\_logic/fuzzy\\_logic\\_introduction.htm](https://www.tutorialspoint.com/fuzzy_logic/fuzzy_logic_introduction.htm)> [Diakses 22 Oktober 2018].
- Ubud Basket, 2018. *Jamur Tiram Organic 200 G (Delivery Wed & Sat) - Ubud Basket*. [Online] Tersedia di: <<https://ubudbasket.com/jamur-tiram-organic-500-g-delivery-saturday/>> [Diakses 31 Oktober 2018].

- Widiastuti, H. & Panji, T., 2008. Pola Aktivitas Enzim Ligninolitik *Pleurotus Ostreatus* pada Limbah Sludge Pabrik Kertas. *Menara Perkebunan*, 1(76), pp. 47-60.
- Zadeh, D. L., 1960. *Logika Fuzzy*. 2 penyunt. U.C. Berkeley: wikipedia.
- Zhao, Q. & Liu, Y., 2012. *Remote Sensing Monitoring System for Agriculture Condition*. Chongqing, China, IEEE.
- Zhu, J., 2018. *Blog - Jingwei Zhu*. [Online] Tersedia di: <<https://jingweizhu.weebly.com/blog/read-temperature-and-humidity-from-dht11-sensor>> [Diakses 31 Oktober 2018].

